**Hercules System/370, ESA/390, z/Architecture Emulator**

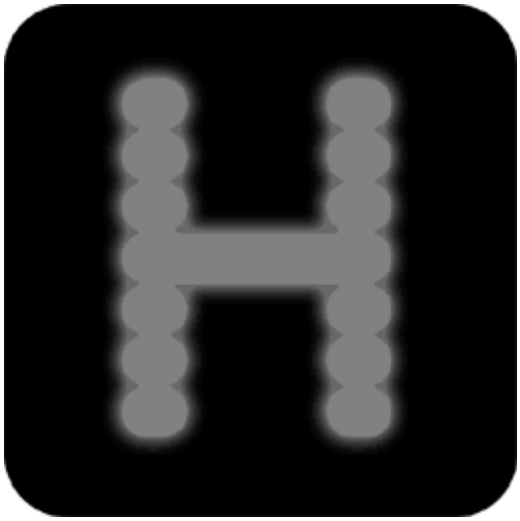
# Hercules – User Reference Guide

***Version 3  Release 11***

**Hercules System/370, ESA/390, z/Architecture Emulator**

# Hercules – User Reference Guide

***Version 3  Release 11***



**Second Edition, November 13, 2015**

**HEUR031100-01**

# Contents

Contents ..... 3
Figures ..... 10
Tables ..... 16
1. Preface ..... 17
1.1 Edition information ..... 17
1.2 What this book is about ..... 17
1.3 Who should read this book ..... 17
1.4 What you need to know to understand this book ..... 17
1.5 How to use this book ..... 17
1.6 Revision Notice ..... 17
1.7 Readers Comments ..... 18
1.8 Legal Advice ..... 18
1.9 Trademarks ..... 18
1.10 Acknowledgements ..... 19
2. Related Publications ..... 20
2.1 Hercules Emulator – General Information ..... 20
2.2 Hercules Emulator – Installation Guide ..... 20
2.3 Hercules Emulator – User Reference Guide ..... 20
2.4 Hercules Emulator – Messages and Codes ..... 20
2.5 Hercules Emulator – Reference Summary ..... 20
3. Hercules Configuration File ..... 21
3.1 The Configuration File ..... 21
3.2 System Parameters ..... 21
3.3 Device Definitions ..... 23
3.4 Coding Rules ..... 24
3.5 Record Format ..... 25
3.6 Sample Configuration File ..... 26
3.7 Symbol Substitution ..... 29
3.8 Enhanced Symbol Substitution ..... 30
4. System Parameter Descriptions ..... 32
4.1 # or * (Comment lines) ..... 32
4.2 ARCHMODE (Initial architecture mode) ..... 33
4.3 ASN_AND_LX_REUSE / ALRF (ESAME ASN and LX REUSE feature) ..... 34
4.4 AUTOMOUNT (Tape automount root directory) ..... 35
4.5 AUTO_SCSI_MOUNT (Automatic SCSI tape mounts) ..... 37
4.6 CCKD (Compressed CKD DASD options) ..... 38
4.7 CNSLPORT (Telnet client port) ..... 43
4.8 CODEPAGE (Codepage conversion table) ..... 44
4.9 CONKPALV (Console and telnet clients keep-alive option) ..... 46
4.10 CPUMODEL (CPU model) ..... 48
4.11 CPUPRIO (CPU thread process priority) ..... 49
4.12 CPUSERIAL (CPU serial number) ..... 50
4.13 CPUVERID (CPU version code) ..... 51
4.14 DEFSYM (Define symbol) ..... 52

4.15 DEVPRIO (Device threads process priority) ..... 53
4.16 DEVTMAX (Maximum number of device threads) ..... 54
4.17 DIAG8CMD (DIAGNOSE 8 command setting) ..... 56
4.18 ECPSVM (ECPS:VM support status (VM)) ..... 58
4.19 ENGINES (Processor engine type) ..... 60
4.20 HERCLOGO (Hercules 3270 logo) ..... 62
4.21 HERCPRIO (Hercules process priority) ..... 63
4.22 HTTPPORT (HTTP server port) ..... 64
4.23 HTTPROOT (HTTP server root directory) ..... 66
4.24 IGNORE (Ignore subsequent INCLUDE errors) ..... 67
4.25 INCLUDE (Include configuration file) ..... 68
4.26 IODELAY (I/O interrupt wait time (LINUX)) ..... 69
4.27 LDMOD (Additional dynamic load modules) ..... 70
4.28 LEGACYSENSEID (SENSE ID CCW (x'E4') feature) ..... 71
4.29 LOADPARM (IPL parameter) ..... 72
4.30 LOGOPT (Log options) ..... 73
4.31 LPARNAME (LPAR name returned by DIAG x'204') ..... 74
4.32 LPARNUM (LPAR identification number) ..... 75
4.33 MAINSIZE (Main storage in MB) ..... 76
4.34 MANUFACTURER (Manufacturer name returned by STSI instruction) ..... 77
4.35 MAXCPU (Maximum number of CPUs) ..... 78
4.36 MODEL (Model names returned by STSI instruction) ..... 80
4.37 MODPATH (Dynamic load module path) ..... 81
4.38 MOUNTED_TAPE_REINIT (Control tape initialization) ..... 82
4.39 NUMCPU (Number of emulated CPUs) ..... 83
4.40 NUMVEC (Number of vector facilities) ..... 84
4.41 OSTAILOR (Intended operating system) ..... 85
4.42 PANRATE (Panel refresh rate) ..... 87
4.43 PANTITLE (Hercules console window title) ..... 88
4.44 PGMPRDOS (LPP license setting) ..... 89
4.45 PLANT (Plant name returned by STSI instruction) ..... 90
4.46 SCLPROOT (SCLP base directory) ..... 91
4.47 SHCMDOPT (Shell command option) ..... 92
4.48 SHRDPORT (Shared device server port) ..... 93
4.49 SYSEPOCH (Base date for TOD clock) ..... 94
4.50 TIMERINT (Internal timer update interval) ..... 95
4.51 TODDRAG (TOD clock drag factor) ..... 96
4.52 TODPRIO (Timer thread process priority) ..... 97
4.53 TRACEOPT (Instruction trace display option) ..... 98
4.54 TZOFFSET (TOD clock offset from GMT) ..... 99
4.55 XPNDSIZE (Expanded storage in MB) ..... 100
4.56 YROFFSET (TOD clock offset from actual date) ..... 101
4.57 Process and Thread Priorities ..... 102
5. Device Definition Descriptions ..... 104
5.1 Local non-SNA 3270 Devices ..... 104
5.2 Integrated 3270 (SYSG) Console ..... 107

5.3 Console Printer-Keyboard Devices ..... 110
5.4 Integrated Console Printer-Keyboard Devices ..... 113
5.5 Card Reader Devices ..... 114
5.6 Card Punch Devices ..... 116
5.7 Line Printer Devices ..... 118
5.8 Emulated Tape Devices ..... 121
5.9 Channel-to-Channel Adapters ..... 135
5.10 FBA DASD Devices ..... 144
5.11 CKD DASD Devices ..... 146
5.12 Communication Lines (Preliminary 2703 BSC Support) ..... 150
6. Hercules Console ..... 153
6.1 Hercules Hardware Console ..... 153
6.2 Web browser interface ..... 155
6.3 Using the keyboard ..... 156
6.4 Log formats ..... 158
6.5 Hercules Console Commands (sorted alphabetically) ..... 161
6.6 Hercules Console Commands (grouped by functionality) ..... 166
7. Console Command Descriptions ..... 172
7.1 !message (SCP priority message) ..... 172
7.2 # or * (Log comment to syslog) ..... 173
7.3 .reply (SCP command) ..... 174
7.4 ? (List all commands / command specific help) ..... 175
7.5 AEA (Display AEA absolute-effective-address tables) ..... 177
7.6 AIA (List AIA absolute-instruction-address fields) ..... 178
7.7 AR (Display access registers) ..... 179
7.8 ARCHMODE (Set architecture mode) ..... 180
7.9 ATTACH (Configure device) ..... 182
7.10 AUTOMOUNT (Show or update allowable tape automount directories) ..... 184
7.11 AUTO_SCSI_MOUNT (Automatic SCSI tape mounts) ..... 186
7.12 B (Set breakpoint) ..... 188
7.13 B+ (Set breakpoint) ..... 189
7.14 B- (Delete breakpoint) ..... 190
7.15 CACHE (Cache command) ..... 191
7.16 CCKD (CCKD command) ..... 193
7.17 CD (Change directory) ..... 199
7.18 CF (Configure CPU online or offline) ..... 200
7.19 CFALL (Configure all CPUs online or offline) ..... 202
7.20 CLOCKS (Display TOD clock and CPU timer) ..... 204
7.21 CMDTGT (Specify the command target) ..... 205
7.22 CONKPALV (Display / alter console TCP/IP keep-alive settings) ..... 207
7.23 CPU (Define target CPU for console displays and commands) ..... 209
7.24 CR (Display or alter control registers) ..... 210
7.25 CSCRIPT (Cancel a running script thread) ..... 212
7.26 CTC (Enable / disable debug packet tracing) ..... 213
7.27 DEFINE (Rename device) ..... 216
7.28 DEFSYM (Define symbol) ..... 217

7.29 DETACH (Remove device) ..... 219
7.30 DEVINIT (Reinitialize device) ..... 220
7.31 DEVLIST (List device or all devices) ..... 222
7.32 DEVTMAX (Display or set maximum device threads) ..... 224
7.33 DS (Display subchannel) ..... 226
7.34 ECPSVM (ECPS:VM commands) ..... 229
7.35 EXIT (Terminate the emulator) ..... 232
7.36 EXT (Generate external interrupt) ..... 234
7.37 F{+/-} (Mark frames usable or unusable) ..... 235
7.38 FPC (Display floating point control register) ..... 236
7.39 FPR (Display floating point registers) ..... 237
7.40 G (Turn off instruction stepping and start CPU) ..... 239
7.41 GPR (Display or alter general purpose registers) ..... 240
7.42 HAO (Hercules Automatic Operator) ..... 242
7.43 HELP (List all commands / command specific help) ..... 245
7.44 HERC (Send Hercules command) ..... 247
7.45 HERCLOGO (Read new logo file) ..... 248
7.46 HST (History of commands) ..... 249
7.47 I (Generate I/O attention interrupt for device) ..... 251
7.48 IODELAY (Display or set I/O delay value) ..... 252
7.49 IPENDING (Display pending interrupts) ..... 254
7.50 IPL (IPL normal from device xxxx) ..... 256
7.51 IPLC (IPL clear from device xxxx) ..... 258
7.52 K (Display CCKD internal trace) ..... 260
7.53 LDMOD (Load a module) ..... 262
7.54 LOADCORE (Load a core image file) ..... 263
7.55 LOADPARM (Set IPL parameter) ..... 264
7.56 LOADTEXT (Load a text deck file) ..... 266
7.57 LOG (Direct log output) ..... 267
7.58 LOGOPT (Change log options) ..... 268
7.59 LPARNAME (Display or set LPAR name) ..... 269
7.60 LPARNUM (Display or set LPAR identification number) ..... 270
7.61 LSDEP (List module dependencies) ..... 272
7.62 LSMOD (List dynamic modules) ..... 273
7.63 MAXRATES (Display maximum observed MIPS/SIO rate or define new reporting interval) ..... 275
7.64 MESSAGE (Display message on console like VM) ..... 277
7.65 MOUNTED_TAPE_REINIT (Control tape initialization) ..... 278
7.66 MSG (Display message on console like VM) ..... 280
7.67 MSGHLD (Display or set timeout of held messages) ..... 281
7.68 MSGNOH (Display message on console like VM, but without header) ..... 283
7.69 OSTAILOR (Specify intended operating system) ..... 284
7.70 PANRATE (Display or set console refresh rate) ..... 286
7.71 PGMTRACE (Trace program interrupts) ..... 288
7.72 PR (Display prefix register) ..... 290
7.73 PSCP (Send system control program priority message) ..... 291
7.74 PSW (Display or alter program status word) ..... 292

7.75 PTT (Set / display pthread trace) ..... 295
7.76 PWD (Print working directory) ..... 299
7.77 QD (Query DASD) ..... 300
7.78 QUIET (Toggle automatic refresh of console display data) ..... 303
7.79 QUIT (Terminate the emulator) ..... 304
7.80 R (Display or alter real storage) ..... 306
7.81 RESTART (Generate restart interrupt) ..... 308
7.82 RESUME (Resume Hercules) ..... 309
7.83 RMMOD (Delete a module) ..... 311
7.84 S (Instruction stepping) ..... 312
7.85 S+ (Instruction stepping on) ..... 314
7.86 S- (Instruction stepping off) ..... 316
7.87 S? (Instruction stepping query) ..... 317
7.88 S{+/-} dev (Turn CCW stepping on or off) ..... 318
7.89 SAVECORE (Save a core image to a file) ..... 321
7.90 SCLPROOT (Set or display SCLP base directory) ..... 323
7.91 SCP (Send system control program command) ..... 325
7.92 SCRIPT (Run a sequence of commands contained in a file) ..... 326
7.93 SCSIMOUNT (Automatic SCSI tape mounts) ..... 327
7.94 SF+ (Create a new shadow file) ..... 329
7.95 SF- (Remove a shadow file) ..... 330
7.96 SFC (Compress a shadow file) ..... 332
7.97 SFD (Display shadow file statistics) ..... 333
7.98 SFK (Perform a chkdsk on the active shadow file) ..... 334
7.99 SH (Shell command) ..... 337
7.100 SHRD (SHRD command) ..... 339
7.101 SIZEOF (Display size of structures) ..... 341
7.102 SSD (Signal shutdown) ..... 342
7.103 START (Start CPU or printer device) ..... 343
7.104 STARTALL (Start all CPUs) ..... 344
7.105 STOP (Stop CPU or printer device) ..... 345
7.106 STOPALL (Stop all CPUs) ..... 346
7.107 STORE (Store CPU status at absolute zero) ..... 347
7.108 SUSPEND (Suspend Hercules) ..... 348
7.109 SYNCIO (Display SYNCIO device statistics) ..... 350
7.110 SYSCLEAR (Issue SYSTEM CLEAR RESET manual operation) ..... 351
7.111 SYSRESET (Issue SYSTEM RESET manual operation) ..... 352
7.112 T (Instruction trace) ..... 353
7.113 T+ (Instruction trace on) ..... 354
7.114 T- (Instruction trace off) ..... 356
7.115 T? (Instruction trace query) ..... 357
7.116 T{+/-} CKD (Turn CKD_KEY tracing on or off) ..... 358
7.117 T{+/-} dev (Turn CCW tracing on or off) ..... 360
7.118 TIMERINT (Display or set timers update interval) ..... 363
7.119 TLB (Display TLB tables) ..... 365
7.120 TODDRAG (Display or set TOD clock drag factor) ..... 366

7.121 TRACEOPT (Instruction trace display options) ..... 368
7.122 TT32 (Control / query CTCI-WIN functionality) ..... 370
7.123 U (Disassemble storage) ..... 372
7.124 UPTIME (Display Hercules Emulator uptime) ..... 374
7.125 V (Display or alter virtual storage) ..... 375
7.126 VERSION (Display version information) ..... 378
8. Hercules Utilities ..... 379
8.1 Overview ..... 379
8.2 CCKDCDSK (CCKD DASD file integrity verification, recovery and repair utility) ..... 382
8.3 CCKDCOMP (CCKD DASD file compression utility) ..... 384
8.4 CCKDDIAG (CCKD DASD file diagnostics utility) ..... 386
8.5 CCKDSWAP (CCKD DASD file swap-endian program) ..... 389
8.6 CKD2CCKD (Copy CKD DASD file to CCKD DASD file) ..... 390
8.7 CCKD2CKD (Copy CCKD DASD file to CKD DASD file) ..... 391
8.8 DASDCAT (Display PDS datasets and members) ..... 392
8.9 DASDCONV (DASD image file conversion program) ..... 396
8.10 DASDCOPY (Copy DASD file to another DASD file) ..... 398
8.11 DASDINIT (DASD image file creation) ..... 400
8.12 DASDISUP (Fix XCTL tables in SVCLIB) ..... 403
8.13 DASDLOAD (DASD loader program) ..... 405
8.14 DASDLS (List datasets on a volume) ..... 413
8.15 DASDPDSU (PDS unload utility) ..... 416
8.16 DASDSEQ (Display sequential datasets) ..... 418
8.17 HETGET (Extract files from an AWS or HET tape file) ..... 420
8.18 HETINIT (Initialize an AWS or HET tape file) ..... 422
8.19 HETMAP (Show information about a HET or AWS tape file) ..... 424
8.20 HETUPD (Update and/or copy an AWS or HET tape file) ..... 429
8.21 TAPECOPY (Copy a SCSI tape to or from an AWSTAPE disk file) ..... 431
8.22 TAPEMAP (Show information about an AWS tape file) ..... 433
8.23 TAPESPLT (Split an AWS tape file) ..... 435
8.24 DMAP2HRC (P/390 DEVMAP conversion program) ..... 437
8.25 FTAPE (SCSI Tape Test Utility) ..... 438
8.26 HercPrt (Remote Hercules Printer Spooler) ..... 446
8.27 PRTPUB (Mainframe print publishing program) ..... 463
8.28 AWSSL (AWS virtual tape) ..... 467
8.29 AWSUTIL (AWS format tape file generation utility) ..... 472
8.30 RAWSTAPE (Reverse AWSTAPE utility) ..... 474
8.31 VTTS (Virtual tape transportation system) ..... 475
9. Shared Device Support ..... 478
9.1 Basics ..... 478
9.2 Caching ..... 478
9.3 Compression ..... 478
9.4 Usage of Shared Devices ..... 479
10. Hercules 3270 Logo ..... 481
10.1 Function ..... 481
10.2 Orders ..... 481

10.3 Variables....483
10.4 Sample....483
11. Starting the Hercules Emulator....486
11.1 Starting Hercules in Native Mode....486
11.2 Starting Hercules with the Windows GUI....487
11.3 Starting Hercules with the Hercules Studio....488
12. The Run-Commands File (hercules.rc)....490
12.1 Function....490
12.2 Run-Commands File Statements....490
12.3 Automating Hercules Startup....492
13. The “Hercules Automatic Operator” (HAO) Facility....495
13.1 HAO Introduction....495
13.2 Defining HAO Rules....495
13.3 Deleting HAO Rules....495
13.4 Substituting Substrings....496
13.5 Limitations....496
13.6 Examples....496
14. Submitting Jobs via the Socket Reader....498
14.1 Socket Reader Basics....498
14.2 Submitting Jobs from Windows....499
14.3 Submitting Jobs from Unix....500
Appendix A: Supported DASD Device Types....502
Appendix B. Configuration of Emulated CPUs....505
B.1 General Explanations and Rules....505
B.1 Examples....505
Appendix C. Syntax....509
C.1 Reading Syntax Descriptions....509
C.2 Reading Syntax Diagrams....510
C.3 Sample Syntax Description....512
C.4 Sample Syntax Diagram....513
Appendix D. Links....515

# Figures

Figure 1: Sample Configuration File ..... 28
Figure 2: OAT File Syntax ..... 142
Figure 3: Hercules Hardware Console ..... 154
Figure 4: Hercules device and status panel ..... 155
Figure 5: Hercules web browser interface ..... 156
Figure 6: Message prefixing using the native Hercules console ..... 159
Figure 7: WinGUI Advanced Logging Options ..... 160
Figure 8: Message prefixing using the console of the Hercules WinGUI ..... 160
Figure 9: Log comment command ..... 173
Figure 10: SCP command ..... 174
Figure 11: “?” command ..... 176
Figure 12: AEA command ..... 177
Figure 13: AIA command ..... 178
Figure 14: AR command ..... 179
Figure 15: ARCHMODE command (display architecture mode) ..... 180
Figure 16: ARCHMODE command (set architecture mode) ..... 181
Figure 17: ATTACH command (configure printer) ..... 182
Figure 18: ATTACH command (configure display terminal) ..... 183
Figure 19: AUTOMOUNT command (add entry) ..... 185
Figure 20: AUTOMOUNT command (list entries) ..... 185
Figure 21: AUTOMOUNT command (delete entry) ..... 185
Figure 22: AUTO_SCSI_MOUNT command (display settings) ..... 187
Figure 23: AUTO_SCSI_MOUNT command (set value) ..... 187
Figure 24: B command ..... 188
Figure 25: B+ command ..... 189
Figure 26: B- command ..... 190
Figure 27: CACHE command ..... 192
Figure 28: CCKD HELP command ..... 197
Figure 29: CCKD OPTS command ..... 197
Figure 30: CCKD STATS command ..... 198
Figure 31: CCKD command (set options) ..... 198
Figure 32: CD command (absolute path) ..... 199
Figure 33: CD command (relative path) ..... 199
Figure 34: CF command (display CPU status) ..... 200
Figure 35: CF command (configure CPU offline) ..... 201
Figure 36: CF command (configure CPU online) ..... 201
Figure 37: CFALL command (display status of all CPUs) ..... 202
Figure 38: CFALL command (configure all CPUs offline) ..... 203
Figure 39: CFALL command (configure all CPUs online) ..... 203
Figure 40: CLOCKS command ..... 204
Figure 41: CMDTGT command (display current setting) ..... 205
Figure 42: CMDTGT command (set target to SCP) ..... 206
Figure 43: CMDTGT command (set target back to Hercules) ..... 206
Figure 44: CONKPALV command ..... 208

Figure 45: CPU command ..... 209
Figure 46: CR command (display control registers) ..... 210
Figure 47: CR command (alter control register) ..... 211
Figure 48: CSCRIPT command ..... 212
Figure 49: CTC command ..... 215
Figure 50: DEFINE command ..... 216
Figure 51: DEFSYM command (list all symbols) ..... 217
Figure 52: DEFSYM command (define new symbol) ..... 218
Figure 53: DEFSYM command (clear defined symbol) ..... 218
Figure 54: DETACH command ..... 219
Figure 55: DEVINIT command ..... 221
Figure 56: DEVLIST command ..... 223
Figure 57: DEVTMAX command (list maximum device threads) ..... 225
Figure 58: DEVTMAX command (set maximum device threads) ..... 225
Figure 59: DS command ..... 226
Figure 60: Subchannel-Information Block ..... 227
Figure 61: Path-Management-Control Word ..... 227
Figure 62: Subchannel-Status Word ..... 228
Figure 63: ECPSVM STATS command ..... 230
Figure 64: ECPSVM ENABLE command ..... 231
Figure 65: ECPSVM LEVEL command ..... 231
Figure 66: EXIT command ..... 233
Figure 67: EXT command ..... 234
Figure 68: F- command ..... 235
Figure 69: F+ command ..... 235
Figure 70: FPC command ..... 236
Figure 71: FPR command (basic format) ..... 237
Figure 72: FPR command (extended format) ..... 238
Figure 73: G command ..... 239
Figure 74: GPR command (display general purpose registers) ..... 240
Figure 75: GPR command (alter general purpose register) ..... 241
Figure 76: HAO command (tgt). ..... 243
Figure 77: HAO command (cmd) ..... 243
Figure 78: HAO command (list) ..... 243
Figure 79: HAO command (del) ..... 244
Figure 80: HAO command (clear) ..... 244
Figure 81: HAO fired command ..... 244
Figure 82: HELP command (information available) ..... 245
Figure 83: HELP command (no information available) ..... 246
Figure 84: HERC command ..... 247
Figure 85: HERCLOGO command ..... 248
Figure 86: HST command ..... 250
Figure 87: I command ..... 251
Figure 88: IODELAY command (display value) ..... 252
Figure 89: IODELAY command (set value) ..... 253
Figure 90: IPENDING command ..... 255

Figure 91: IPL command ..... 257
Figure 92: IPLC command ..... 259
Figure 93: K command ..... 261
Figure 94: LDMOD command ..... 262
Figure 95: LOADCORE command ..... 263
Figure 96: LOADPARM command (display IPL parameter) ..... 264
Figure 97: LOADPARM command (set IPL parameter) ..... 265
Figure 98: LOG command ..... 267
Figure 99: LOGOPT command ..... 268
Figure 100: LPARNAME command (display LPAR name) ..... 269
Figure 101: LPARNAME command (define LPAR name) ..... 269
Figure 102: LPARNUM command (set LPARNUM). ..... 270
Figure 103: LPARNUM command (display LPARNUM). ..... 271
Figure 104: LSDEP command ..... 272
Figure 105: LSMOD command ..... 274
Figure 106: MAXRATES command (set the interval time) ..... 275
Figure 107: MAXRATES command (display maximum rates) ..... 276
Figure 108: MESSAGE command ..... 277
Figure 109: MOUNTED_TAPE_REINIT command (display settings). ..... 279
Figure 110: MOUNTED_TAPE_REINIT command (disallow tape mount reinitialization) ..... 279
Figure 111: MSG command ..... 280
Figure 112: MSGHELD command (set new interval) ..... 281
Figure 113: MSGHELD command (release messages) ..... 282
Figure 114: MSGNOH command ..... 283
Figure 115: OSTAILOR command (display intended operating system) ..... 285
Figure 116: OSTAILOR command (specify intended operating system) ..... 285
Figure 117: PANRATE command (list current value) ..... 286
Figure 118: PANRATE command (set new value) ..... 287
Figure 119: PGMTRACE command (display settings) ..... 288
Figure 120: PGMTRACE command (change settings) ..... 289
Figure 121: PR command ..... 290
Figure 122: PSW command (display PSW). ..... 293
Figure 123: PSW command (modify PSW) ..... 293
Figure 124: PSW command (modify PSW) ..... 294
Figure 125: PTT command (display trace options / set options and start trace) ..... 296
Figure 126: PTT command (display trace entries) ..... 298
Figure 127: PTT command (start trace and issue automatic display) ..... 298
Figure 128: PWD command ..... 299
Figure 129: QD command ..... 301
Figure 130: QUIET command ..... 303
Figure 131: QUIT command ..... 305
Figure 132: R command (display real storage) ..... 307
Figure 133: R command (alter real storage) ..... 307
Figure 134: RESTART command ..... 308
Figure 135: RESUME command ..... 310
Figure 136: RMMOD command ..... 311

Figure 137: S command ..... 313
Figure 138: S+ command ..... 315
Figure 139: S- command ..... 316
Figure 140: S? command ..... 317
Figure 141: S+ dev command ..... 319
Figure 142: S- dev command ..... 320
Figure 143: SAVECORE command (save specific area) ..... 322
Figure 144: SAVECORE command (save from begin to end) ..... 322
Figure 145: SCLPROOT command (display directory) ..... 323
Figure 146: SCLPROOT command (set directory) ..... 324
Figure 147: SCP command ..... 325
Figure 148: SCRIPT command ..... 326
Figure 149: SCSIMOUNT command ..... 328
Figure 150: SF+ command ..... 329
Figure 151: SF- command ..... 331
Figure 152: SF- NOMERGE command ..... 331
Figure 153: SFC command ..... 332
Figure 154: SFD command ..... 333
Figure 155: SFK command (chkdsk level 2) ..... 335
Figure 156: SFK command (chkdsk level 4) ..... 336
Figure 157: SH command ..... 338
Figure 158: SHRD command (initialize trace table) ..... 339
Figure 159: SHRD command (print trace table) ..... 340
Figure 160: SIZEOF command ..... 341
Figure 161: SSD command ..... 342
Figure 162: START command (start printer) ..... 343
Figure 163: START command (start CPU) ..... 343
Figure 164: STARTALL command ..... 344
Figure 165: STOP command (stop printer) ..... 345
Figure 166: STOP command (stop CPU) ..... 345
Figure 167: STOPALL command ..... 346
Figure 168: STORE command ..... 347
Figure 169: SUSPEND command ..... 349
Figure 170: SYNCIO command ..... 350
Figure 171: SYSCLEAR command ..... 351
Figure 172: SYSRESET command ..... 352
Figure 173: T command ..... 353
Figure 174: T+ command ..... 355
Figure 175: T- command ..... 356
Figure 176: T? command ..... 357
Figure 177: T+CKD command ..... 359
Figure 178: T-CKD command ..... 359
Figure 179: T+ dev command ..... 361
Figure 180: T- dev command ..... 362
Figure 181: T+ dev command (several devices) ..... 362
Figure 182: TIMERINT command (list current value) ..... 363

Figure 183: TIMERINT command (set new value) ..... 364
Figure 184: TLB command ..... 365
Figure 185: TODDRAG command (display TOD clock factor) ..... 366
Figure 186: TODDRAG command (set TOD clock factor) ..... 367
Figure 187: TRACEOPT command (TRADITIONAL) ..... 369
Figure 188: TRACEOPT command (REGSFIRST) ..... 369
Figure 189: TRACEOPT command (NOREGS) ..... 369
Figure 190: TT32 DEBUG command ..... 370
Figure 191: TT32 NODEBUG command ..... 371
Figure 192: TT32 STATS command ..... 371
Figure 193: U command ..... 373
Figure 194: UPTIME command (uptime < 1 day). ..... 374
Figure 195: UPTIME command (uptime > 1 week). ..... 374
Figure 196: V command (display virtual storage with length) ..... 376
Figure 197: V command (alter virtual storage) ..... 376
Figure 198: V command (display virtual storage with range) ..... 377
Figure 199: V command (display virtual storage with ‘S’ option) ..... 377
Figure 200: VERSION command ..... 378
Figure 201: CCKDCDSK utility output ..... 383
Figure 202: CCKDCOMP utility output ..... 385
Figure 203: CCKDDIAG utility output ..... 388
Figure 204: CCKDSWAP utility output ..... 389
Figure 205: DASDCAT utility output (create PDS member list) ..... 393
Figure 206: DASDCAT utility output (list a member from a PDS) ..... 394
Figure 207: DASDCAT utility output (list all members of a PDS) ..... 395
Figure 208: DASDCONV utility output ..... 397
Figure 209: DASDCOPY utility output ..... 399
Figure 210: DASDINIT utility output ..... 402
Figure 211: DASDISUP utility output ..... 404
Figure 212: DASDLOAD utility output (XMIT method, message level 1) ..... 409
Figure 213: DASDLOAD utility output (XMIT method, message level 3) ..... 410
Figure 214: DASDLOAD utility output (SEQ method, message level 5) ..... 412
Figure 215: DASDLS utility output (basic information) ..... 414
Figure 216: DASDLS utility output (extended information) ..... 415
Figure 217: DASDPDSU utility output ..... 417
Figure 218: DASDSEQ utility output ..... 419
Figure 219: HETGET utility output ..... 421
Figure 220: HETINIT utility output ..... 423
Figure 221: HETMAP utility output (print all label and file information) ..... 427
Figure 222: HETMAP utility output (print only dataset information) ..... 427
Figure 223: HETMAP utility output (TAPEMAP compatible output) ..... 428
Figure 224: HETUPD utility output ..... 430
Figure 225: TAPECOPY utility output ..... 432
Figure 226: TAPEMAP utility output (map AWS tape file) ..... 434
Figure 227: TAPESPLT utility output ..... 436
Figure 228: FTAPE utility output (display status) ..... 445

Figure 229: HercPrt Program Options Page ..... 447
Figure 230: HercPrt PDF Options Page ..... 449
Figure 231: Job Seperator Control File Sample ..... 461
Figure 232: HercPrt Report Sample ..... 462
Figure 233: AWSSL JCL (Create AWS virtual tape) ..... 470
Figure 234: AWSSL JCL (Create datasets from AWS virtual tape) ..... 471
Figure 235: AWSUTIL utility JCL ..... 473
Figure 236: RAWSTAPE utility JCL ..... 474
Figure 237: IEBGENER JCL for RAWSTAPE utility ..... 474
Figure 238: VTT2DISK utility JCL. ..... 476
Figure 239: VTT2TAPE utility JCL ..... 476
Figure 240: VTT2CNVU utility JCL ..... 477
Figure 241: Logo File ..... 484
Figure 242: Logo Screen ..... 485
Figure 243: Run-commands (.rc) file ..... 493
Figure 244: Batch file to start tn3270 sessions ..... 494
Figure 245: HercRdr Help Screen ..... 499
Figure 246: SUBmit REXX for SPF/PC ..... 500
Figure 247: The Hercsub Perl Script ..... 501
Figure 248: Sample Syntax Description ..... 513
Figure 249: Sample Syntax Diagram ..... 514

# Tables

Table 1: Hercules System Parameters ..... 23
Table 2: Hercules Device Definitions ..... 24
Table 3: Supported codepage mappings ..... 45
Table 4: Process Priority Conversions ..... 102
Table 5: Thread Priority Conversions ..... 103
Table 6: Default CU Types ..... 148
Table 7: Normal cursor handling ..... 157
Table 8: Extended cursor handling ..... 158
Table 9: Message prefix overview ..... 158
Table 10: Hercules Console Commands (sorted alphabetically) ..... 165
Table 11: Hercules Console Commands (grouped by functionality) ..... 171
Table 12: Sense ID ..... 301
Table 13: Read Device Characteristics ..... 302
Table 14: Read Configuration Data ..... 302
Table 15: DASD image file maintenance utilites ..... 379
Table 16: TAPE file maintenance utilities ..... 380
Table 17: Miscellaneous utilities ..... 380
Table 18: Third party utilities ..... 380
Table 19: Mainframe utilities ..... 381
Table 20: Sample Control Files ..... 452
Table 21: HercPrt PDF and RTF Options ..... 456
Table 22: HercPrt ASA Control Characters and ASCII Equivalents ..... 459
Table 23: Supported CKD DASD Devices ..... 503
Table 24: Supported FBA DASD Devices ..... 504
Table 25: Correct CPU configuration (example 1) ..... 506
Table 26: Correct CPU configuration (example 2) ..... 506
Table 27: Correct CPU configuration (example 3) ..... 507
Table 28: Correct CPU configuration (example 4) ..... 507
Table 29: Correct CPU configuration (example 5) ..... 508
Table 30: Incorrect CPU configuration (example 6) ..... 508
Table 31: Reading Syntax Descriptions ..... 509
Table 32: Reading Syntax Diagrams ..... 512

# 1. Preface

## 1.1 Edition information

This edition applies to the Hercules S/370, ESA/390 and z/Architecture Emulator, Release 3.11.0 and to all subsequent versions, releases and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of software you are using.

## 1.2 What this book is about

This book is a guide for using and operating the Hercules Emulator and related additional products (both required and optional). For guidance in installation or debugging Hercules or for a general overview, additional manuals are available.

Please note that some information can be found in more than one manual. This redundancy is not intended to unnecessarily expand the manuals, rather to help find all necessary information in one place.

## 1.3 Who should read this book

This book is mainly intended for people who are responsible for operating the Hercules Emulator. It may also be useful if you are responsible for installing the Hercules Emulator.

## 1.4 What you need to know to understand this book

To understand this book, you should be familiar using software under the Linux, Windows, Linux and/or Mac OS X operating systems. You should also have experience with Linux command shells or native DOS (Microsoft Disk Operating System) and the Microsoft Windows command shell.

Last but not least you should be familiar with IBM mainframe environments (hardware and software) and the underlying ideas and concepts as Hercules emulates IBM mainframe hardware.

## 1.5 How to use this book

This book is designed as a reference for all aspects the Hercules Emulator and related products. You can go through the book chapter by chapter or you can use the book as a reference for all questions regarding the operation of Hercules.

## 1.6 Revision Notice

| | |
|---|---|
| Hercules Release: | Version 3 Release 11 Modification 0 |
| Publication Number: | HEUR031100 |
| SoftCopy Name: | HerculesUserReference |
| Revision Number: | HEUR031100-01 |
| Date: | November 13, 2015 |

## 1.7 Readers Comments

If you like or dislike anything of this book please send a mail or email to the address below. Feel free to comment any errors or lack of clarity. Please limit your comments on the information in this specific book and also include the “Revision Notice” just above. Thank you for your help.

Send your comments by email to the Hercules-390 discussion group:

hercules-390@yahoogroups.com

## 1.8 Legal Advice

Hercules implements only the raw S/370, ESA/390, and z/Architecture instruction set, it does not provide any operating system facilities. This means that you need to provide an operating system or standalone program which Hercules can load from an emulated disk or tape device. You will have to write the operating system or standalone program yourself unless you possess a license from IBM to run one of their operating systems on your PC or use IBM programs and operating systems which have been placed in the public domain.

***NOTE: It is YOUR responsibility to comply with the terms of the license for the operating system you intend to run on the Hercules Emulator.***

## 1.9 Trademarks

The following is a list of trademark acknowledgments and copyright notices for product and company names mentioned in this book. Other product and company names in this book that are not listed below may be the trademarks or registered trademarks of their respective owners.

- IBM, System/370, ESA/390, z/Architecture, MVS, OS/390, z/OS, VM, VM/ESA, z/VM, VSE, VSE/ESA, z/VSE are trademarks or registered trademarks of International Business Machines Corporation (IBM).
- Windows XP, Windows Vista, Windows 7, Windows 8, Windows Server 2003, Windows Server 2008, Windows Server 2012, Visual C++ Toolkit, Visual C++ Express are trademarks of Microsoft Corporation.
- Linux is a trademark owned by Linus Torvalds. The Linux Mark Institute is the exclusive licensor of the Linux trademark on behalf of its owner Linus Torvalds.
- WinPcap is copyrighted by NetGroup, Politecnico di Torino (Italy).
- Cygwin is copyrighted by Red Hat, Inc.
- Vista tn3270 is copyrighted by Tom Brennan Software.
- Pentium, XEON are trademarks or registered trademarks of Intel Corporation.
- Athlon, Opteron are trademarks or registered trademarks of Advanced Micro Devices (AMD), Inc.
- Xmit Manager is copyrighted by Neal Johnston-Ward.
- FLEX-ES is a registered trademark of Fundamental Software, Inc.
- UMX Virtual Mainframe is a registered trademark of UMX Technologies.

## 1.10 Acknowledgements

The Hercules manuals would not have been possible without the assistance of many people and I would like to thank all those who helped me. In particular I would like to thank:

- The Hercules developers for their documentation on various websites from which I derived a great deal of information.
- Roger Bowler and Fish for proof-reading the manuals.
- Loris Degoianni for allowing me to use parts of the original WinPcap documentation.
- Tom Brennan for allowing me to use parts of his Vista tn3270 documentation.
- My colleagues for working with early previews of the documentation, beginning with just a few pages.
- Mike Cairns for reviewing and editing the manuals.
- Robert Allan for providing the “Linux Installation” part.
- Lutz Mader for providing the “Mac OS X Installation” part.

If anyone feels they have been forgotten on this list please let me know.

Peter Glanzmann

# 2. Related Publications

## 2.1 Hercules Emulator – General Information

The Hercules “General Information” manual provides you an overview of the ideas and concepts of the Hercules Emulator as well as a documentation of the emulator’s functionality. It explains what Hercules does and does not do. It helps you decide if the software fits your needs fulfills your requirements.

## 2.2 Hercules Emulator – Installation Guide

The Hercules "Installation Guide" shows you how to install Hercules and all related optional and required software components under the Microsoft Windows, Linux and Apple MacIntosh OS X operating systems.

After going through the installation guide you will have a working emulator environment ready to IPL a S370, S/390 or z/Architecture mainframe operating system.

## 2.3 Hercules Emulator – User Reference Guide

The Hercules “User Reference” leads you through all aspects of the emulator’s operation. It provides instruction in the operation of the Hercules Emulator with and without the Windows GUI. The usage details for all Hercules utilities are also covered in this guide.

After reading this manual you should be able to work with Hercules and the Hercules console, create virtual devices, understand backup/restore procedures and general housekeeping within the Hercules environment.

## 2.4 Hercules Emulator – Messages and Codes

The “Messages and Codes” manual provides a detailed explanation of all Hercules related messages. It is the primary source for troubleshooting and debugging when experience problems with Hercules.

## 2.5 Hercules Emulator – Reference Summary

The Hercules "Reference Summary" booklet lists all the system parameters, device definitions, console commands, Hercules utilities etc. along with their arguments.

This booklet is intended as a quick reference guide for experienced users. Consult the Hercules "User Reference Guide" for more detailed and additional information.

# 3. Hercules Configuration File

## 3.1 The Configuration File

This chapter describes the configuration file for the Hercules Emulator. By default the configuration file is named “hercules.cnf” or “hercules.conf”. It is located in the configuration directory, refer to the Hercules Installation Guide for details of the directory structure. The configuration file contains the processor definitions in the system parameters section, the device layout in the device definitions section and Hercules runtime parameters also in the system parameters section.

The configuration file - especially the device definition part - is roughly equivalent to the IOCDS definitions found on a real mainframe environment. It is an ASCII text file that is read and interpreted by the Hercules Emulator during its initialization phase.

## 3.2 System Parameters

The system parameters describe the processor definition and some Hercules internal runtime parameters.

The processor definitions include the processor model, CPU serial number, memory configuration and architecture mode. They describe the processor hardware that is to be emulated.

The Hercules runtime parameters define values that Hercules requires as an emulated environment. These parameters include values like port numbers, directory paths and priorities.

The following table shows an overview of all valid system parameters:

| System Parameter | Description |
|---|---|
| # | Comment line |
| * | Comment line |
| ARCHMODE | Initial architecture mode |
| ASN_AND_LX_REUSE / ALRF | ESAME ASN and LX REUSE feature |
| AUTOMOUNT | Tape automount root directory |
| AUTO_SCSI_MOUNT | Automatic SCSI tape mounts |
| CCKD | Compressed CKD DASD options |
| CNSLPORT | Telnet client port |
| CODEPAGE | Codepage conversion table |
| CONKPALV | Console and telnet clients keep-alive option |
| CPUMODEL | CPU model |
| CPUPRIO | CPU thread process priority |

| System Parameter | Description |
|---|---|
| CPUSERIAL | CPU serial number |
| CPUVERID | CPU version code |
| DEFSYM | Define symbol |
| DEVPRIO | Device threads process priority |
| DEVTMAX | Maximum number of device threads |
| DIAG8CMD | DIAGNOSE 8 command setting |
| ECPSVM | ECPS:VM support status (VM) |
| ENGINES | Processor engine type |
| HERCLOGO | Hercules 3270 logo |
| HERCPRIO | Hercules process priority |
| HTTPPORT | HTTP server port |
| HTTPROOT | HTTP server root directory |
| IGNORE | Ignore subsequent INCLUDE errors |
| INCLUDE | Include configuration file |
| IODELAY | I/O interrupt wait time (LINUX) |
| LDMOD | Additional dynamic load modules |
| LEGACYSENSEID | SENSE ID CCW (x'E4') feature |
| LOADPARM | IPL parameter |
| LOGOPT | Log options |
| LPARNAME | LPAR name returned by DIAG x'204' |
| LPARNUM | LPAR identification number |
| MAINSIZE | Main storage in MB |
| MANUFACTURER | Manufacturer name returned by STSI instruction |
| MAXCPU | Maximum number of CPUs |
| MODEL | Model names returned by STSI instruction |
| MODPATH | Dynamic load module path |

| System Parameter | Description |
|---|---|
| MOUNTED_TAPE_REINIT | Control tape initialization |
| NUMCPU | Number of emulated CPUs |
| NUMVEC | Number of vector facilities |
| OSTAILOR | Intended operating system |
| PANRATE | Panel refresh rate |
| PANTITLE | Hercules console window title |
| PGMPRDOS | LPP license setting |
| PLANT | Plant name returned by STSI instruction |
| SCLPROOT | SCLP base directory |
| SHCMDOPT | Shell command option |
| SHRDPORT | Shared device server port |
| SYSEPOCH | Base date for TOD clock |
| TIMERINT | Internal timer update interval |
| TODDRAG | TOD clock drag factor |
| TODPRIO | Timer thread process priority |
| TRACEOPT | Instruction trace display option |
| TZOFFSET | TOD clock offset from GMT |
| XPNDSIZE | Expanded storage in MB |
| YROFFSET | TOD clock offset from actual date |

**Table 1: Hercules System Parameters**

## 3.3 Device Definitions

The device definitions describe the layout of the emulated hardware i.e: DASD, tape, terminals and printers. It is comparable with the IOCDS found in a real mainframe environment. Each emulated device must be specified in this part of the configuration file.

The following table shows an overview of all valid devices:

| Device Type | Device | Emulated by |
|---|---|---|
| 3270, 3278 | Local non-SNA 3270 display or printer | TN3270 client connection |
| SYSG | Integrated 3270 (SYSG) console | TN3270 client connection |
| 1052, 3215 | Console printer-keyboards | Telnet client connection |
| 1052-C, 3215-C | Integrated console printer-keyboards | Integrated on Hercules console |
| 1442, 2501, 3505 | Card readers | Disk file(s), ASCII or EBCDIC |
| 3525 | Card punch | Disk file, ASCII or EBCDIC |
| 1403, 3211 | Line printers | Disk file, ASCII |
| 3410, 3420, 3422, 3430, 3480, 3490, 3590, 9347, 8809 | Tape drives | Disk file, CD-ROM or SCSI tape |
| 3088 | Channel-to-Channel Adapter | “CTCT” driver |
| (( CTCI )) | Channel-to-Channel link to host TCP/IP stack | “CTCI” TUN/TAP driver |
| (( LCS )) | IBM 2216 router,<br>IBM 3172 running ICP,<br>IBM 8232 LCS device,<br>LCS3172 driver of a P/390,<br>IBM Open Systems Adapter (OSA) | “LCS” (LAN channel station)<br><br>TUN/TAP driver |
| 3310, 3370, 9313, 9332, 9335, 9336, 0671 | FBA direct access storage devices | Disk file |
| 2305, 2311, 2314, 3330, 3340, 3350, 3375, 3380, 3390, 9345 | CKD direct access storage devices | Disk file |
| 2703 | Communication line | TCP socket |

**Table 2: Hercules Device Definitions**

## 3.4 Coding Rules

There are only a few rules for creating configuration files. The file must be an ASCII text file. Blank lines, or lines beginning with a hash (“#”) sign or an asterisk (“*”), are treated as comments and are not checked.

The system parameters may appear in any order but must precede all device definitions. Each system parameter must be specified on a separate line.

The device records may also appear in any order but must follow the system parameters. There must be one device definition for each I/O device or for each group of identical I/O devices.

## 3.5 Record Format

The following sections describe the format of the Hercules configuration file records. The format is slightly different between system parameters and device definitions.

### *3.5.1 Format of the system parameters*

The record format of the system parameters is as follows:

***system-parameter argument***

where:

***system-parameter*** is the name of the system parameter.

***argument*** is the value assigned to the system parameter.

Examples:

```
CPUMODEL   3090
HTTPROOT   /usr/local/share/hercules/
PANRATE    FAST
```

### *3.5.2 Format of the device definitions*

The format of device definition is as follows:

***devnum(s) devtype [argument(s)]***

where:

***devnum(s)*** is either: a single devnum (see details below), a range of devnums separated by a dash (“-“), a count of devnums seperated by a dot (“.”), or a comma (“,”) separated list of devnums.

Examples include:

- 0200-0210
- 0300.10
- 0400,0410
- 0100,0110-011F

If devnums specifies more than one device then all devices will have identical characteristics. All devices defined as a group must be defined on a single channel. A channel is defined as a continuous group of

256 (or hexadecimal 100) devices. For example devices 0010 and 0020 are on the same channel, whereas devices 0100 and 0210 are not on the same channel.

The devnum itself is either a 1 to 4 digit hexadecimal number in the range 0000 to FFFF for ESA/390 or 0000 to 0FFF for S/370. The device number uniquely identifies each device to the operating system.

***devtype*** is the device type.

***argument(s)*** is a list of parameters depending on the device type. These parameters are explained in the sections that describe each device type.

Examples:

```
0120        3380  mvsv5r.120
0230.16     3270  GROUP1 192.168.100.0 255.255.255.0
0583-0587   3420  *  maxsizeM=170  eotmargin=131072
```

## 3.6 Sample Configuration File

The following figure shows a real example of a Hercules configuration file, used for running the MVS 3.8J operating system. Please note, that not all possible parameter and definitions are contained in this sample.

```
#
# Hercules Emulator Control file...
# Description: MVS V3.8J
# MaxShutdownSecs: 60
#

#
# System parameters
#

CPUMODEL         7490                       # CPU model
CPUSERIAL        001963                     # CPU serial number
CPUVERID         FD                         # CPU version code
LPARNAME         HERCULES                   # LPAR name returned by DIAG X'204'
MODEL            EMULATOR                   # Model name returned by STSI
PLANT            ZZ                         # Plant name returned by STSI
MANUFACTURER     HRC                        # Manufacturer name returned by STSI
MAINSIZE         16                         # Main storage in MB
XPNDSIZE         0                          # Expanded storage in MB
CODEPAGE         DEFAULT                    # Codepage conversion table
CNSLPORT         3270                       # TELNET client port
NUMCPU           2                          # Number of emulated CPU's
NUMVEC           0                          # Number of vector facilities
ARCHMODE         S/370                      # Initial architecture mode
LOADPARM         0148....                   # IPL parameters
SYSEPOCH         1928                       # Base date for TOD clock
```

```
TZOFFSET         +0000                      # TOD clock offset from GMT
OSTAILOR         OS/390                     # Intended operating system
PANRATE          1000                       # Panel refresh rate in ms
PANTITLE         MVS_V3.8J                  # Console window panel title
LOGOPT           TIMESTAMP                  # Hercules log options
CONKPALV         (3,1,10)                   # Console and telnet clients keep-alive option
DEVTMAX          0                          # Maximum number of device threads
PGMPRDOS         LICENSED                   # Do not emulate IFL hardware
SHCMDOPT         NODIAG8                    # SHELL command behaviour
DIAG8CMD         DISABLE                    # DIAGNOSE 8 command setting
ASN_AND_LX_REUSE DISABLE                    # ESAME 'ASN and LX REUSE' optional feature
AUTO_SCSI_MOUNT  NO                         # Automatic detection of SCSI tape mounts
IODELAY          0                          # I/O interrupt wait time (Linux only)
ECPSVM           NO                         # ECPS:VM support status (VM only)
HERCPRIO         0                          # Hercules process priority
CPUPRIO          0                          # CPU thread process priority (default: 15)
DEVPRIO          0                          # Device threads process priority (default: 8)
TODPRIO          -20                        # Timer thread process priority
TIMERINT         DEFAULT                    # Internal timers update interval
HTTPROOT         D:/Hercules/html           # HTTP server root directory
HTTPPORT         8081 AUTH p390 fzrx6v      # HTTP server port / authorization required
SHRDPORT         3990                       # Shared devcice server port
DEFSYM           DASDPATH D:/MVS/DASD       # Define path for DASD devices

#
# Device Definitions
#

# Card Readers

000C   3505   D:/MVS/JCL/DUMMY.JCL eof ascii trunc

# Card Punches

000D   3525   D:/MVS/PCH/pch00d.txt ascii crlf

# Line Printers

000E   1403   D:/MVS/PRT/prt00e.txt crlf
000F   1403   D:/MVS/PRT/prt00f.txt crlf
030E   1403   D:/MVS/LOG/hardcopy.log crlf

# Display Terminals

0010   3270   *   192.168.0.100
0011   3270   *   192.168.0.100
00C0   3270   *   192.168.0.100
00C1   3270   *   192.168.0.100
00C2   3270   *   192.168.0.100
00C3   3270   *   192.168.0.200
00C4   3270   *   192.168.0.200
00C5   3270   *   192.168.0.200
00C6   3270   *
```

```
00C7   3270   *

# DASD Devices

0130   2314   D:/MVS/DASD/SORT00.CCKD
0131   2314   D:/MVS/DASD/SORT01.CCKD
0132   2314   D:/MVS/DASD/SORT02.CCKD
0133   2314   D:/MVS/DASD/SORT03.CCKD
0134   2314   D:/MVS/DASD/SORT04.CCKD
0135   2314   D:/MVS/DASD/SORT05.CCKD
0140   3350   D:/MVS/DASD/WORK00.CCKD
0141   3350   D:/MVS/DASD/WORK01.CCKD
0142   3350   D:/MVS/DASD/PRD000.CCKD
0143   3350   D:/MVS/DASD/PRD001.CCKD
0144   3350   D:/MVS/DASD/PRD002.CCKD
0145   3350   D:/MVS/DASD/PRD003.CCKD
0148   3350   D:/MVS/DASD/MVSRES.CCKD
0149   3350   D:/MVS/DASD/MVSDLB.CCKD
0160   3340   D:/MVS/DASD/PAGE00.CCKD
0161   3340   D:/MVS/DASD/PAGE01.CCKD
0240   3350   D:/MVS/DASD/PUB000.CCKD
0241   3350   D:/MVS/DASD/PUB001.CCKD
0242   3350   D:/MVS/DASD/PUB002.CCKD
0243   3350   D:/MVS/DASD/PUB003.CCKD
0248   3350   D:/MVS/DASD/SYS000.CCKD
0249   3350   D:/MVS/DASD/SYS001.CCKD
024A   3350   D:/MVS/DASD/SYS002.CCKD
024B   3350   D:/MVS/DASD/SYS003.CCKD
0340   3350   D:/MVS/DASD/CBT000.CCKD
0341   3350   D:/MVS/DASD/CBT001.CCKD
0342   3350   D:/MVS/DASD/CBT002.CCKD
0343   3350   D:/MVS/DASD/CBT003.CCKD
0344   3350   D:/MVS/DASD/SPOOL0.CCKD
0345   3350   D:/MVS/DASD/SPOOL1.CCKD
0348   3350   D:/MVS/DASD/TST000.CCKD
0349   3350   D:/MVS/DASD/TST001.CCKD
034A   3350   D:/MVS/DASD/TST002.CCKD
034B   3350   D:/MVS/DASD/TST003.CCKD

# TAPE Devices

0480   3420   *   MAXSIZEM=210   EOTMARGIN=131072
0481   3420   *   MAXSIZEM=210   EOTMARGIN=131072
0482   3420   *   MAXSIZEM=210   EOTMARGIN=131072
0483   3420   *   MAXSIZEM=210   EOTMARGIN=131072
```

**Figure 1: Sample Configuration File**

## 3.7 Symbol Substitution

In configuration and device statements, as well as in console commands and OAT files, symbols may be substituted for text.

### *3.7.1 Syntax*

To substitute symbol *symbol* with its contents the symbol should be enclosed within parenthesis and preceded by a $ sign. For example, if symbol *FOO* contains the text string *"BAR"* then *$(FOO)* will be substituted with the string *"BAR"*;. Symbol names are case sensitive.

### *3.7.2 Examples*

```
...
DEFSYM TAPEDIR "/home/hercules/tapes"
...
0380 3420 $(TAPEDIR)/scratch.aws
...
```

In the above example device 0380 will be a 3420 loaded with the AWSTAPE file in the directory "/home/hercules/tapes/scratch.aws".

### *3.7.3 Special symbols*

#### 3.7.3.1 Device group symbols

When multiple devices are defined with a single device definition, then the following symbols may be used. These are defined as containing the relevant device address for each device:

- CUU 3 digit device number, upper case hexadecimal digits
- CUUU 4 digit device number, upper case hexadecimal digits
- cuu 3 digit device number, lower case hexadecimal digits
- cuuu 4 digit device number, lower case hexadecimal digits

Example:

```
0200,0201 3340 /home/hercules/dasds/myvols.$(CUU)
```

will define 2 3340 packs with device 0200 being loaded from the file “myvols.200” and device 0201 defined from “myvols.201”.

#### 3.7.3.2 Environment variables

If a symbol is not explicitly defined by a DEFSYM statement and an environment variable by the same name exists, then the string contents of that environment variable will be used for substitution.

#### 3.7.3.3 Undefined symbols

If a symbol is not defined by an explicit DEFSYM, is not an automatically generated symbol or is not an environment variable, an empty string will be substituted.

### *3.7.4 Escaping substitution, recursion*

To specify the '$' string without incurring substitution an additional '$' sign should be used. For example, $$(FOO) will not be substituted. If substitution is required but the preceding text is to contain a '$' sign as the very last character then $$$(FOO) would be specified. Thus if symbol FOO contains "BAR", then $$(FOO) will remain "$$(FOO)" while $$$(FOO) will become "$BAR". Substitution is not recursive, only one substitution pass is made.

## 3.8 Enhanced Symbol Substitution

Enhanced symbol substitution is an additional method for specifiying symbols. It differs from the above normal symbol substitution in several important ways.

### *3.8.1 Differences from conventional Symbol Substitution*

The syntax of the enhanced symbol substitution differs from conventional symbol substitution. Enhanced substitution symbol names are specified using "${var}" (dollar sign and brace) rather than "$(var)" (dollar sign and parenthesis).

This enhanced syntax supports the specification of a default value that will be used whenever the named symbol is not otherwise defined. The default value is placed within the opening and closing braces just as the symbol name is but separated from it by either a single equal sign "=" or a colon-equal-sign ":=". For example, specifying "${DASD_PATH=dasd/}" in your configuration file will result in the value of the "DASD_PATH" symbol or environment variable being substituted, or if the variable is undefined, the value "dasd/" instead. If no default value is specified then an empty string will be used instead.

An important distinction of enhanced symbol substitution is that it only occurs from host defined environment variables and not from any identically named DEFSYM symbol should one exist. For example, if environment variable "FOO" is defined with the value "bar", then the configuration file statement "DEFSYM FOO myfoo", followed immediately by the statement "${FOO}" causes the value "bar" to be substituted and not "myfoo" as might be expected. Whereas the statement "$(FOO)", since it is a normal symbol substitution sequence, will be substituted with "myfoo", the value defined by the preceding DEFSYM statement.

To clarify, the two symbol substitution techniques are supported completely separately from each other. DEFSYM allows you to use previously defined or undefined, private (internally defined), symbols separate from the host operating system's environment variable pool. Enhanced symbol substitution allows read-only access to the host environment variable pool with no support for modifying an already defined symbol (environment variable), however with a convenient means of defining a default value to be used should the specified host environment variable be currently undefined.

It is important to note that symbol names, potentially being the names of environment variables, are subject to whatever case sensitivity the host operating system happens to enforce or allow. Under Windows environment variables are not case sensitive, whereas on other operating systems they may be. Thus "${FOO}", "${foo}", "${Foo}" etc, all cause the same value to be substituted on Windows, whereas the DEFSYM symbols $(FOO) and $(foo), could be substituted with two completely different values under a case sensitive operating system. IE: the Hercules implementation of DEFSYM is case sensitive across all supported platforms.

### 3.8.2 Syntax

To substitute symbol *symbol* with the current environment variable value, the symbol should be enclosed within braces and preceded by a $ sign. For example, if an environment variable named FOO holds the value "BAR", then ${FOO} will be substituted with the string "BAR". If the environment variable "FOO" is not defined then a null (empty) string is substituted instead.

If the string "${FOO:=myfoo}" is used instead, then the value "BAR" will still be substituted if the value "BAR" was previously assigned to FOO, but will be substituted with the value "myfoo" instead if the environment variable FOO is currently undefined.

Note too that the default value is a literal string and no substitution is applied to it. Thus attempting to use the syntax "${foo=${bar}}" will not yield the expected results. It will not be substituted with the currently defined value of the "bar" environment variable, but rather by the literal string "${bar" followed immediately by the literal character '}'.

Symbol names (environment variable names) are not case sensitive on Windows whereas they might be on other host operating systems.

### 3.8.3 Examples

```
...
0120  3390  ${DASD_PATH=dasd/}  mvsv5r.120
...
```

In the above example, device 0120 will be a 3390 loaded with the DASD file from the directory specified in the environment variable "DASD_PATH" or, if the DASD-PATH environment variable is not defined, the value "dasd/" instead.

# 4. System Parameter Descriptions

## 4.1 # or * (Comment lines)

### *4.1.1 Function*

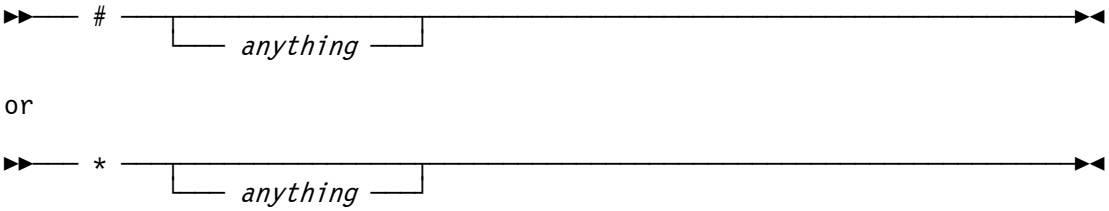
The hash (“#”) or asterisk ("*") symbols mark the beginning of a comment or a full comment line.

### *4.1.2 Syntax*

**Descriptive**

```
# [anything]
or
* [anything]
```

**Diagram**




### *4.1.3 Parameter*

**anything** Any text following the hash or asterisk symbol.

### *4.1.4 Examples*

**Example 1:**

Add comment lines to the configuration file and add comment after the system parameter.

```
************************************
* System Parameters                *
************************************

NUMCPU   2      # Number of emulated CPUs
MAINSIZE 1024   # Main storage in MB
```

## 4.2 ARCHMODE (Initial architecture mode)

### *4.2.1 Function*

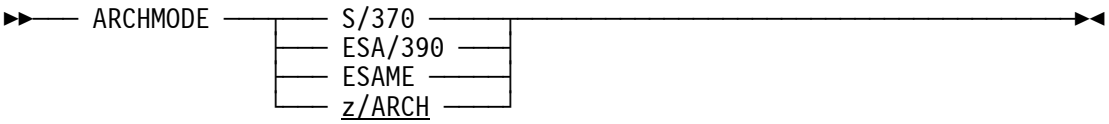
The archmode parameter specifies the initial architecture mode.

### *4.2.2 Syntax*

**Descriptive**

```
ARCHMODE {S/370 | ESA/390 | ESAME | z/ARCH}
```

**Diagram**




### *4.2.3 Parameter*

**S/370** Use S/370 for OS/360, VM/370 and MVS 3.8

**ESA/390** Use ESA/390 for MVS/XA, MVS/ESA, OS/390, VM/ESA, VSE/ESA, Linux/390 and ZZSA. z/OS can be run until version 1.2 with ESA/390 mode without installed bimodal feature or until version 1.4 if the bimodal feature is installed.

**ESAME** Use ESAME (Enterprise System Architecture, Modal Extensions) for z/OS, z/VM, z/VSE and z/Linux. When ESAME is specified, the machine will always be IPL'ed in ESA/390 mode but the system is capable of being switched into the z/Architecture mode after IPL. This is handled automatically by all z/Architecture operating systems.

**z/ARCH** This is the same as ESAME.

### *4.2.4 Examples*

**Example 1:**

Specify ESA/390 architecture mode.

```
ARCHMODE ESA/390
```

## 4.3 ASN_AND_LX_REUSE / ALRF (ESAME ASN and LX REUSE feature)

### *4.3.1 Function*

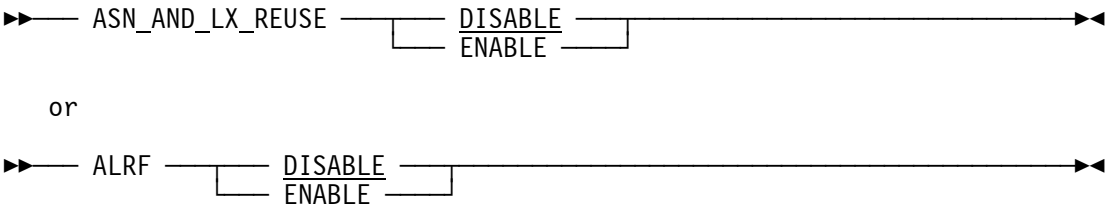
This parameter specifies if the “ASN and LX Reuse Facility” is to be disabled (the default) or enabled. The ASN_AND_LX_REUSE parameter can be abbreviated as ALRF.

### *4.3.2 Syntax*

**Descriptive**

```
ASN_AND_LX_REUSE {DISABLE | ENABLE}
   or
ALRF {DISABLE | ENABLE}
```

**Diagram**




### *4.3.3 Parameter*

**DISABLE** The “ASN and LX Reuse Facility” is disabled. This is the default and is always disabled for S/370 and ESA/390 modes. Set it to DISABLE if the operating system does not support this feature and it inadvertently sets CR0 bit 44 to 1, usually leading to an unexpected program interrupt when instructions such as LASP are issued.

**ENABLE** The “ASN and LX Reuse Facility” is enabled. Set this parameter to ENABLE if the operating system supports this z/Architecture feature and the use of this feature is desired.

### *4.3.4 Examples*

**Example 1:**

Enable the ASN and LX Reuse Facility.

```
ASN_AND_LX_REUSE ENABLE
```

## 4.4 AUTOMOUNT (Tape automount root directory)

### *4.4.1 Function*

Specifies the host system directory where the guest is allowed or not allowed to automatically load virtual tape volumes from. Prefix allowable directories with a '+' (plus) sign and unallowable directories with a '-' (minus) sign. The default prefix if neither is specified is the '+' sign (an allowable directory).

All host system virtual tape volumes to be automounted by the guest must reside within one of the specified allowable host system directories or any of its subdirectories while not also being within any of the specified unallowable directories or any of their subdirectories, in order for the guest-invoked automount to be accepted.

Specifying a disallowed automount directory does not preclude the Hercules operator from manually mounting any file via the DEVINIT console command - even one in a currently defined disallowed automount directory. The AUTOMOUNT statement only controls guest-invoked automatic tape mounts and not manual tape mounts performed by the Hercules operator.

All directories must be specified on separate statements, but as many statements as needed may be specified in order to describe the desired allowable/unallowable directories layout. For convenience, an AUTOMOUNT console command is also provided to dynamically add new or remove existing automount directories at any time.

The automount feature is activated whenever you specify at least one allowable or unallowable directory. If only unallowable directories are specified, then the current directory becomes the only defined allowable automount directory by default.

All specified directories are always resolved to fully-qualified absolute directory paths before being saved.

**Caution:** Enabling this feature may have security consequences depending on which allowable host system directories you specify as well as how your guest operating system enforces authorized use of the Set Diagnose (X'4B') channel command code.

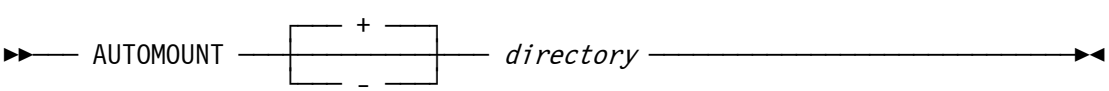
Refer to the description of the virtual tape device 'NOAUTOMOUNT' option for more information.

### *4.4.2 Syntax*

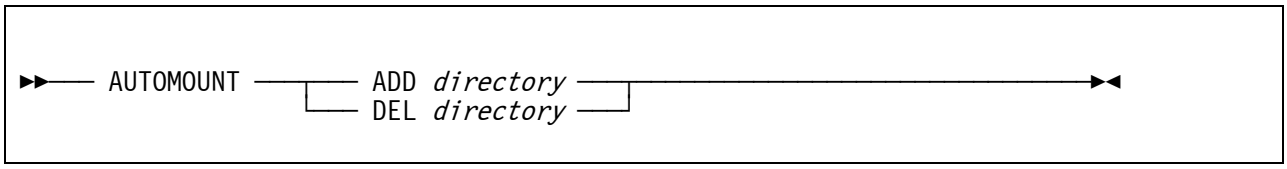
**Descriptive**

```
AUTOMOUNT [ + | - ]directory
   or
AUTOMOUNT {ADD directory | DEL directory}
```

**Diagram**




or




### 4.4.3 Parameter

**ADD or +** Add an entry to the list of allowable tape automount directories.

**DEL or -** Delete an entry from the list of allowable tape automount directories.

**directory** Specifies the host system directory where the guest is allowed or not to automatically load virtual tape volumes from.

### 4.4.4 Examples

**Example 1:**

Specify directory “D:\MVS\TAPE” as the host system directory from where the guest is allowed to automatically load virtual tape volumes.

```
AUTOMOUNT +D:/MVS/TAPE
   or
AUTOMAOUNT ADD D:/MVS/TAPE
```

## 4.5 AUTO_SCSI_MOUNT (Automatic SCSI tape mounts)

### *4.5.1 Function*

The AUTO_SCSI_MOUNT parameter specifies whether automatic detection of SCSI tape mounts are enabled or not.

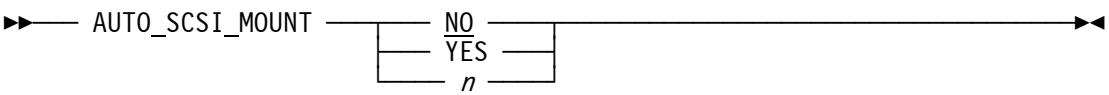
**NOTE!** Enabling this option may negatively impact Hercules performance depending on how the host operating system (Windows, Linux etc.) handles SCSI attached tape drive status queries.

### *4.5.2 Syntax*

**Descriptive**

```
AUTO_SCSI_MOUNT {NO | YES | n}
```

**Diagram**




### *4.5.3 Parameter*

**NO** NO indicates that the SCSIMOUNT option is disabled, forcing all SCSI tape mounts to be done manually via an appropriate DEVINIT command.

**YES** Yes enables the option and causes periodic queries of the SCSI tape drive in a five second interval to automatically detect when a tape is mounted. YES is equivalent with “SCSIMOUNT 5”.

**n** A value from 1 to 99 seconds (inclusive) enables this option and causes periodic queries of the SCSI tape drive to automatically detect if a tape is mounted.

### *4.5.4 Examples*

**Example 1:**

Enable automatic detection of SCSI tape mounts and set the periodic queries of the SCSI tape drive to a 30 seconds interval.

```
AUTO_SCSI_MOUNT 30
```

## 4.6 CCKD (Compressed CKD DASD options)

### *4.6.1 Function*

The CCKD configuration statement is used to alter CCKD processing. The CCKD configuration statement supports the same options as the CCKD console command.

### *4.6.2 Syntax*

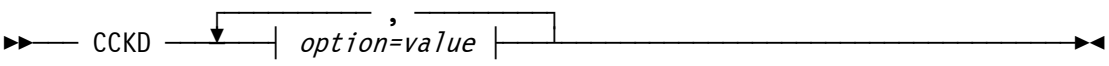
**Descriptive**

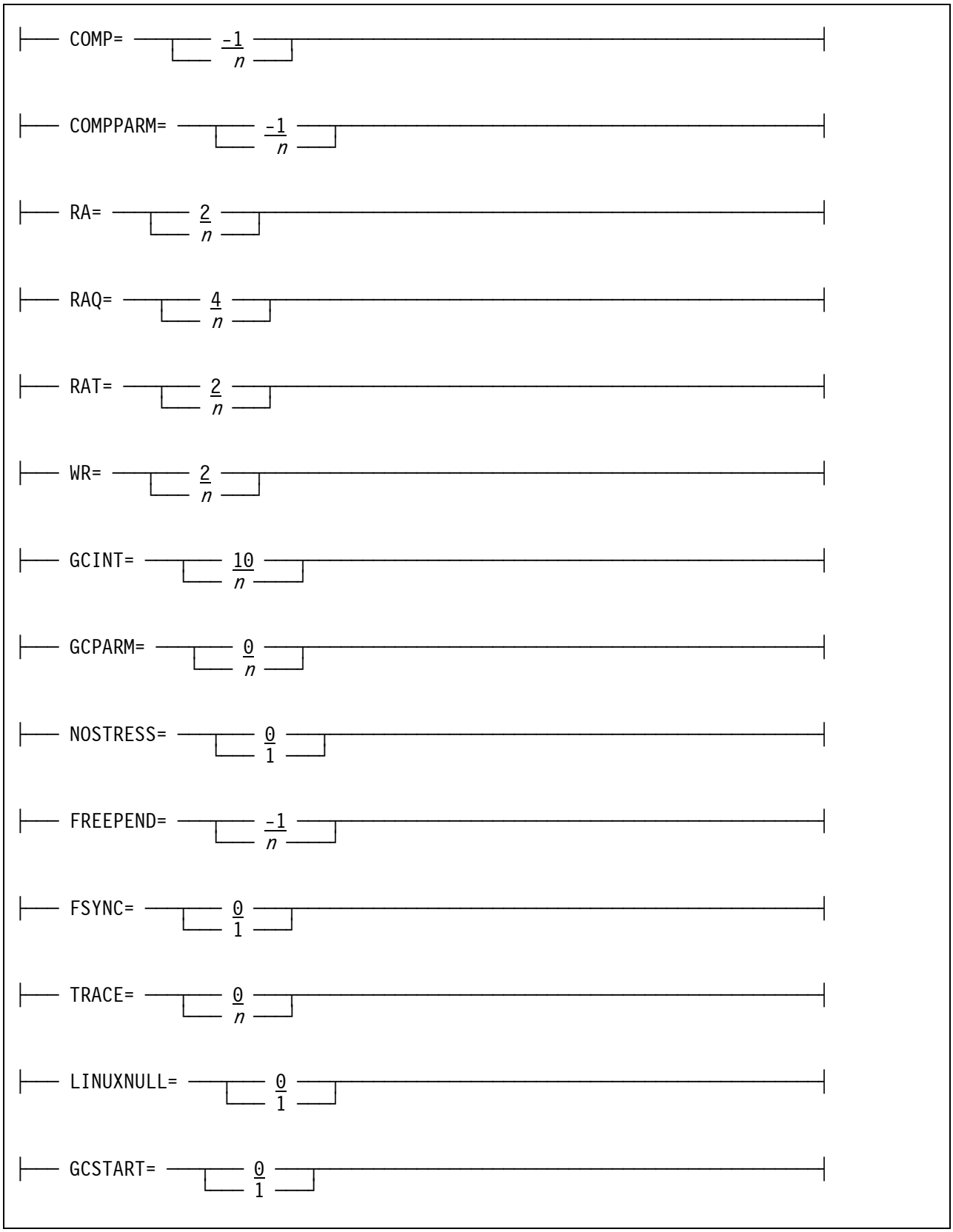
CCKD *option=value* [,*option=value* [,*option=value* ... ]]

where option can be:

[COMP={-1 | *n*}]
[,COMPPARM={-1 | *n*}]
[,RA={2 | *n*}]
[,RAQ={4 | *n*}]
[,RAT={2 | *n*}]
[,WR={2 | *n*}]
[,GCINT={10 | *n*}]
[,GCPARM={0 | *n*}]
[,NOSTRESS={0 | 1}]
[,FREEPEND={-1 | *n*}]
[,FSYNC={0 | 1}]
[,TRACE={0 | *n*}]
[,LINUXNULL={0 | 1}]
[,GCSTART={0 | 1}]

**Diagram**




COMP=
-1
n
COMPPARM=
-1
n
RA=
2
n
RAQ=
4
n
RAT=
2
n
WR=
2
n
GCINT=
10
n
GCPARM=
0
n
NOSTRESS=
0
1
FREEPEND=
-1
n
FSYNC=
0
1
TRACE=
0
n
LINUXNULL=
0
1
GCSTART=
0
1

### *4.6.3 Parameter*

**option** Set a CCKD option. Multiple options may be specified, separated by commas with no intervening blanks.

**The CCKD options are:**

**COMP=n** Specifies the compression type to be used. This overrides the compression used for all CCKD files. The default (-1) means don't override the compression. Valid compression types are:

**-1** Default

**0** None

**1** zlib

**2** bzip2

**COMPPARM=n** Overrides the compression parameter. A higher value generally means more compression at the expense of CPU and/or storage. The default (-1) means don't override the compression parameter. The value of *n* can be from -1 and 9.

**RA=n** Sets the Number of read ahead threads. When sequential track or block group access is detected, some number (RAT=n) of tracks or block groups are queued (RAQ=n) to be read by one of the read ahead threads. The default is 2, the value of *n* can be a number from 1 to 9.

**RAQ=n** Sets the size of the read ahead queue. When sequential track or block group access is detected, some number (RAT=n) of tracks or block groups are queued in the read ahead queue. The default is 4, the value of *n* can be a number from 0 to 16. A value of zero disables read ahead.

**RAT=n** Sets the number of tracks or block groups to read ahead when sequential track or block group access is detected. The default is 2, the value of ratn can be a number from 0 to 16. A value of zero disables read ahead.

**WR=n** Sets the number of writer threads. When the cache is flushed, updated cache entries are marked write pending and a writer thread is signalled. The writer thread compresses the track or block group and writes the compressed image to the emulation file.

A writer thread is CPU-intensive while compressing the track or block group and I/O-intensive while writing the compressed image. The writer thread runs one nicer than the CPU thread(s). The default is 2, a value from 1 to 9 can be specified

**GCINT=n** This is the number of seconds the garbage collector thread waits during an interval. At the end of an interval the garbage collector performs space recovery, flushes the cache and optionally 'fsyncs' the emulation file. However, the file will not be 'fsynced' unless at least 5 seconds have elapsed since the last synchronization (FSYNC). The default is 10 seconds. You can specify a number between 1 and 60.

**GCPARM=n** A value affecting the amount of data moved during the garbage collectors space recovery routine. The garbage collector determines an amount of space to move based on the ratio of free space to used space in an emulation file and on the number of free spaces in the file. The garbage collector wants to reduce the free space to used space ratio and the number of free spaces.

The value is logarithmic; a value of 8 means moving $2^8$ the selected value while a negative value similarly decreases the amount to be moved. Normally, 256K will be moved for a file in an interval. Specifying a value of 8 can increase the amount to 64M. At least 64K will be moved. Specifying a large value (such as 8) may not increase the garbage collection efficiency correspondingly. The default is 0. You can specify a number from -8 to 8.

**NOSTRESS=n** Indicates whether stress writes will occur or not. A track or block group may be written under stress when a high percentage of the cache is pending write or when a device I/O thread is waiting for a cache entry. When a stressed write occurs, the compression algorithm and/or compression parm may be relaxed, resulting in faster compression but usually a larger compressed image. If NOSTRESS is set to one, then a stressed situation is ignored. You would typically set this value to one when you want create the smallest emulation file possible in exchange for a possible performance degradation. The default is 0. You can specify 0 (enable stressed writes) or 1 (disable stressed writes).

**FREEPEND=n** Specifies the free pending value for freed space. When a track or block group image is written, the space it previously occupied is freed. This space will not be available for future allocations until *n* garbage collection intervals have completed. In the event of a catastrophic failure, previously written track or block group images should be recoverable if the current image has not yet been written to the physical disk. By default the value is set to -1 which means that if FSYNC is specified then the value is 1 otherwise it is 2. If 0 is specified then freed space is immediately available for new allocations. The default is -1. You can specify a number from -1 to 4.

**FSYNC=n** Enables or disables FSYNC. When FSYNC is enabled then the disk emulation file is synchronized with the physical hard disk at the end of a garbage collection interval (no more often than 5 seconds though). This means that if FREEPEND is non-zero and a catastrophic error occurs, the emulated disks should be recovered coherently. However, FSYNC may cause performance degradation depending on the host operating system and / or the host operating system level. The default is 0 (fsync disabled), you can specify 0 (disable FSYNC) or 1 (enable FSYNC).

**TRACE=n** value Specifies the number of CCKD trace entries. You would normally specify a non-zero when debugging or capturing a problem in CCKD code. When the problem occurs, you should enter the "k" Hercules console command which will print the trace table entries. The default is 0. You can specify a number between 0 and 200000. Each trace entry represents 128 bytes. Normally, for debugging, it is recommended to use 100000.

**LINUXNULL=n** If set to 1 then tracks written to 3390 CCKD volumes that were initialized with the *-linux* option will be checked if they are null (that is if all 12 4096 byte user records contain zeroes). This is used by the DASDCOPY utility. The default is 0.

**GCSTART=n** If set to 1 then space recovery will become active on any emulated disks that have free space. Normally space recovery will ignore emulated disks until they have been updated. The default is 0.

**Notes:**

- *raq* should be at least as large as *ra*. Read ahead threads are scheduled from entries in the read ahead queue. Likewise *rat* should not exceed *raq* because only *raq* tracks or block groups can be queued at any time.

- The number of writer threads *wr* should usually be 1 more than the number of host processors. This is because one writer thread could be CPU-bound (compressing a track or block-group image) and the other could be i/o-bound (writing the compressed image).

- The garbage collection interval governs the maximum time in seconds an updated track or block group image will reside in storage before being written to the emulation file. A large value may mean more data loss if a catastrophic error occurs. A small value may mean that more CPU time is spent compressing images.

  For example, suppose that a particular image is updated several times each second. If the interval is changed from the default 5 seconds to 1 second, then that image will be compressed and written 5 times more frequently. A large value may cause more cache flushes within a garbage collection interval. These kinds of flushes mean that a read will wait because there are no avail-

able cache entries, slowing the emulated operating system. A large value will also cause more pending free space to build up (since free space is flushed each interval). This may mean that the garbage collector space recovery routine will perform more work and the resulting emulation file may be larger.

- Specify *fsync=1* and *gcint=5* if you are seriously concerned about your data being lost due to a failure. *fsync* will ensure your data on disk is coherent. However, fsync may cause a noticeable performance degradation. Note that an fsync will not be performed more often than every 5 seconds.

### *4.6.4 Examples*

**Example 1:**

Set the CCKD options to use bzip2 as compression method and using maximum compression.

```
CCKD COMP=2,COMPPARM=9
```

**Example 2:**

Set the CCKD options to use the default compression method, as well as default compression. Disable stressed writes and set the number of trace entries to 100000.

```
CCKD COMP=-1,COMPPARM=-1,NOSTRESS=1,TRACE=100000
```

## 4.7 CNSLPORT (Telnet client port)

### *4.7.1 Function*

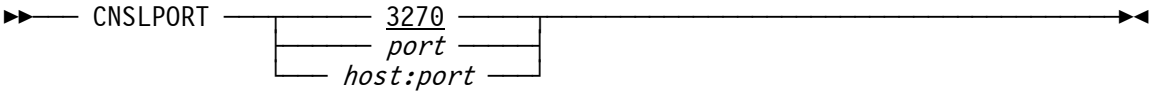
The CNSLPORT parameter specifies the port number (in decimal), on which the telnet server will listen. The statement may also have the form host:port, where the telnet console server will bind to the specified address.

### *4.7.2 Syntax*

**Descriptive**

```
CNSLPORT {3270 | port | host:port}
```

**Diagram**




### *4.7.3 Parameter*

**host** The IP address of the host to which the telnet server will bind to. If an IP address is given then it must be a valid IP address for the host system.

**port** The port number (decimal) on which the telnet server will listen. The port number must not be in use by any other server. Ports below 1024 cannot be used unless Hercules is running as root or is otherwise authorized to use low ports.

### *4.7.4 Examples*

**Example 1:**

Set the port number on which the telnet server will listen to 3270.

```
CNSLPORT 3270
```

**Example 2:**

Specify 192.168.1.10 as the IP address of the host to which the telnet server will bind to and set the port number on which the telnet server will listen to 3270.

```
CNSLPORT 192.168.1.10:3270
```

## 4.8 CODEPAGE (Codepage conversion table)

### *4.8.1 Function*

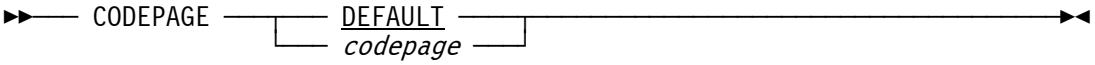
This parameter specifies the codepage conversion table used for the ASCII / EBCDIC translation. If no codepage is given the environment variable HERCULES_CP will be inspected. The default codepage used is “DEFAULT”.

### *4.8.2 Syntax*

**Descriptive**

```
CODEPAGE {DEFAULT | codepage}
```

**Diagram**




### *4.8.3 Parameter*

**codepage** Specifies the codepage conversion table used for ASCII / EBCDIC translation. Supported codepage mappings are shown in the table below. Iconv single byte codepages may also be used (e.g. “UTF8/EBCDIC-CP-NL”).

**DEFAULT** DEFAULT specifies the traditional Hercules codepage.

Supported codepage mappings:

| Mapping | ASCII | EBCDIC |
|---|---|---|
| 437/037 | 437 PC United States | 037 United States/Canada |
| 437/500 | 437 PC United States | 500 Latin 1 |
| 437/1047 | 437 PC United States | 1047 Open Systems Latin 1 |
| 819/037 | 819 ISO-8859-1 Latin 1 | 037 United States/Canada |
| 819/037v2 | 819 ISO-8859-1 Latin 1 | 037 United States/Canada SHARE |
| 819/273 | 819 ISO-8859-1 Latin 1 | 273 CECP Austria/Germany |
| 819/277 | 819 ISO-8859-1 Latin 1 | 277 CECP Denmark/Norway |
| 819/278 | 819 ISO-8859-1 Latin 1 | 278 CECP Finland/Sweden |
| 819/280 | 819 ISO-8859-1 Latin 1 | 280 CECP Italy |
| 819/284 | 819 ISO-8859-1 Latin 1 | 284 CECP Spain |

| Mapping | ASCII | EBCDIC |
|---|---|---|
| 819/285 | 819 ISO-8859-1 Latin 1 | 285 CECP United Kingdom |
| 819/297 | 819 ISO-8859-1 Latin 1 | 297 CECP France |
| 819/500 | 819 ISO-8859-1 Latin 1 | 500 CECP International |
| 819/1047 | 819 ISO-8859-1 Latin 1 | 1047 Open Systems Latin 1 |
| 850/273 | 850 PC Latin 1 | 273 Austria/Germany |
| 850/1047 | 850 PC Latin 1 | 1047 Open Systems Latin 1 |
| 1252/037 | 1252 Windows Latin 1 | 037 United States/Canada |
| 1252/037v2 | 1252 Windows Latin 1 | 037 United States/Canada SHARE |
| 1252/1047 | 1252 Windows Latin 1 | 1047 Open Systems Latin 1 |
| 1252/1140 | 1252 Windows Latin 1 | 1140 United States/Canada with Euro sign |

**Table 3: Supported codepage mappings**

### *4.8.4 Examples*

**Example 1:**

Set the codepage conversion table to the default.

```
CODEPAGE DEFAULT
```

**Example 2:**

Set the codepage conversion table to 437/500.

```
CODEPAGE 437/500
```

## 4.9 CONKPALV (Console and telnet clients keep-alive option)

### *4.9.1 Function*

The CONKPALV parameter specifies the tn3270 console and telnet clients keepalive option values that control automatic detection of disconnected tn3270/telnet client sessions.

This is a built-in feature of TCP/IP and allows detection of unresponsive TCP/IP connections and not idle clients. That is to say, your connection will not be terminated after 3 seconds of idle time. Your 3270 session can remain idle for many minutes or hours or days without any data being transmitted. If the TCP/IP stack at the other end of the connection (not your 3270 client itself) fails to respond to the internal keepalive probe packets however, then it means that the TCP/IP stack itself is down or there has been a physical break in the connection.

Thus, even if your 3270 client is completely idle, your system's TCP/IP stack itself should still respond to the keepalive probes sent by the TCP/IP stack at the Hercules end of the link. If it doesn't, then TCP/IP will terminate the tn3270/telnet session which will cause Hercules to disconnect the terminal.

The three values can also be modified on-demand via the conkpalv panel command, which has the exact same syntax. Note that the syntax is very unforgiving: no spaces are allowed anywhere within the parentheses and each value must be separated from the other with a single comma.

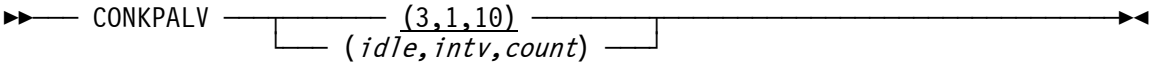
Please also note that not all systems support being able to modify all three values. That is, not all values may be able to be changed, and it is system dependent which values you can change and which values you cannot. On Windows for example, the *count* value is ignored and cannot be changed from its default value of 10. Other systems may ignore one or more or all three values and use platform defaults instead. This is entirely system dependent. Check you system's documentation for details regarding which values can be changed and which cannot as well as how to adjust your system's default values.

### *4.9.2 Syntax*

**Descriptive**

```
CONKPALV {(3,1,10) | (idle,intv,count)}
```

**Diagram**




### *4.9.3 Parameter*

**idle** The *idle* value specifies the number of seconds of inactivity until the first keep-alive probe is sent. The default for the *idle* value is 3 seconds.

**intv** The *intv* value specifies the interval in seconds between the probes if no acknowledgement is received from the previous probe. The default for *intv* is 1 second.

**count** The *count* value specifies the number of unacknowledged keep-alive packets sent before the connection is considered to have failed. The default value is 9 for non-Windows platforms and 10 for Windows systems.

*Note:* On Windows platforms the count value is ignored and cannot be changed from its default value of 10.

### *4.9.4 Examples*

**Example 1:**

Set the tn3270 console and telnet clients keep-alive option settings on a Unix platform to the following values: 5 seconds of inactivity until the first keep-alive probe is sent, 3 seconds for the interval between the probes and to 15 unacknowledged keep-alive packets sent before the connection is considered to have failed.

```
CONKPALV (5,3,15)
```

## 4.10 CPUMODEL (CPU model)

### *4.10.1 Function*

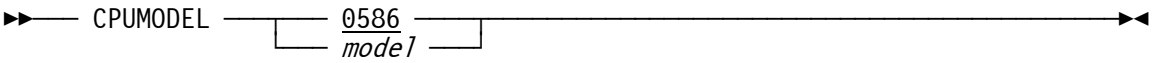
The CPUMODEL parameter specifies the 4 hexadecimal digit CPU model number stored by the STIDP instruction.

### *4.10.2 Syntax*

**Descriptive**

```
CPUMODEL {0586 | model}
```

**Diagram**




### *4.10.3 Parameter*

**model** Any valid 4 digit hexadecimal CPU model number. A list of the valid model numbers can be found in the Hercules Windows GUI file “cpu-types.txt”. The default (“0586”) is not a mainframe CPU model number, but is related to the underlying PC architecture.

### *4.10.4 Examples*

**Example 1:**

Specify a 7490 CPU model.

```
CPUMODEL 7490
```

**Example 2:**

Specify a 3090 CPU model.

```
CPUMODEL 3090
```

## 4.11 CPUPRIO (CPU thread process priority)

### *4.11.1 Function*

CPUPRIO specifies the priority of the CPU thread. Default is a nice value of 15, which means a low priority such that I/O can be scheduled and completed in favour of CPU cycles. On multi-CPU systems a real CPU can be “dedicated” to Hercules by giving the CPU-thread a very high dispatching priority (-20).

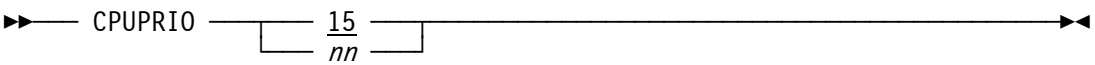
**Caution:** CPUPRIO should not have a higher dispatching priority than the TOD clock and timer thread.

### *4.11.2 Syntax*

**Descriptive**

```
CPUPRIO {15 | nn}
```

**Diagram**




### *4.11.3 Parameter*

**nn** This value specifies the priority for the CPU thread. For details on the priority values see section 4.57 (“Process and Thread Priorities”). The default is 15.

### *4.11.4 Examples*

**Example 1:**

Give the CPU-thread a very high dispatching priority of -20.

```
CPUPRIO -20
```

## 4.12 CPUSERIAL (CPU serial number)

### *4.12.1 Function*

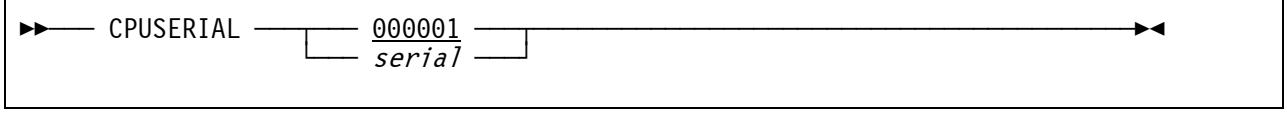
CPUSERIAL specifies the 6 hexadecimal digit CPU serial number stored by the STIDP instruction.

### *4.12.2 Syntax*

**Descriptive**

```
CPUSERIAL {000001 | serial}
```

**Diagram**




### *4.12.3 Parameter*

**serial** Any valid 6 digit hexadecimal CPU serial number. The default serial number is 000001.

### *4.12.4 Examples*

**Example 1:**

Set the CPU serial number to 001963.

```
CPUSERIAL 001963
```

## 4.13 CPUVERID (CPU version code)

### *4.13.1 Function*

CPUVERID specifies the 2 hexadecimal digit CPU version code stored by the STIDP instruction. The default version code is “FD” when ARCHMODE S/370 or ARCHMODE ESA/390 is specified. For the z/ARCH (or ESAME architecture mode respectively) the version code is alwyas stored as “00” and any value specified here is ignored.

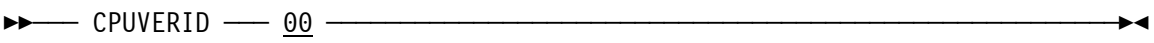
### *4.13.2 Syntax*

**Descriptive**

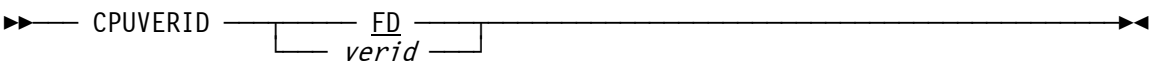
```
CPUVERID 00                    (For z/ARCH and ESAME)
CPUVERID {FD | verid}          (For S/370 and ESA/390)
```

**Diagram**

For z/ARCH and ESAME:




For S/370 and ESA/390:




### *4.13.3 Parameter*

**verid** Any valid 2 digit hexadecimal CPU version code. A list of valid version codes can be found in the Hercules Windows GUI file “cpu-types.txt”.

### *4.13.4 Examples*

**Example 1:**

Set the CPU version code to FD.

```
CPUVERID FD
```

## 4.14 DEFSYM (Define symbol)

### *4.14.1 Function*

Defines symbol *symbol* is to contain value *value*. The symbol can then be the object of a substitution used later in the configuration file or for console commands. If *value* contains blanks or spaces it must be enclosed within quotes or apostrophes. See chapter “Symbol Substitutions” for a more in-depth discussion on this feature.

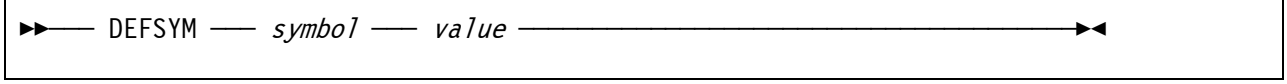
Substitution is available in configuration statements, meaning it is possible to perform substitution in the *DEFSYM* statement itself. However, symbols are always defined as the last step in the process, so attempting to self define a symbol will result in an empty string.

### *4.14.2 Syntax*

**Descriptive**

```
DEFSYM symbol value
```

**Diagram**




### *4.14.3 Parameter*

**symbol** The name of a symbol.

**value** The value that is assigned to the symbol.

### *4.14.4 Examples*

**Example 1:**

Define a symbol “TAPEDIR” with the value “/home/hercules/tapes”.

```
DEFSYM TAPEDIR "/home/hercules/tapes"
```

## 4.15 DEVPRIO (Device threads process priority)

### *4.15.1 Function*

This value specifies the priority of the device threads.

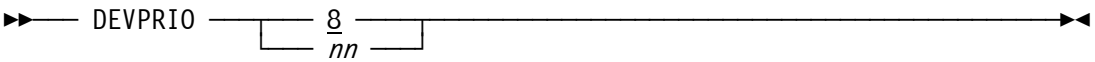
**Caution:** DEVPRIO should not have a higher dispatching priority than the TOD clock and timer thread.

### *4.15.2 Syntax*

**Descriptive**

```
DEVPRIO {8 | nn}
```

**Diagram**




### *4.15.3 Parameter*

**nn** This value specifies the priority for the device threads. For details on the priority values see section 4.57 (“Process and Thread Priorities”). The default is 8.

### *4.15.4 Examples*

**Example 1:**

Set a priority of 8 for the device threads.

```
DEVPRIO 8
```

## 4.16 DEVTMAX (Maximum number of device threads)

### *4.16.1 Function*

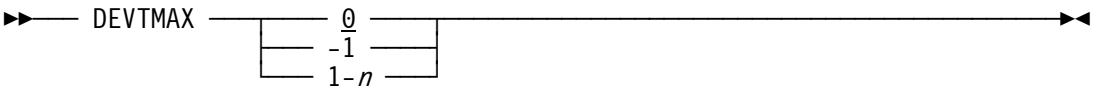
DEVTMAX specifies the maximum number of device threads allowed.

### *4.16.2 Syntax*

**Descriptive**

DEVTMAX {0 | -1 | 1-*n*}

**Diagram**




### *4.16.3 Parameter*

**0** Specify 0 to cause an unlimited number of 'semi-permanent' threads to be created on an 'as-needed' basis. This is the default. With this option, a thread is created to service an I/O request for a device if one does not already exist. When the I/O is complete the thread enters an idle state waiting for new work. If a new I/O request for the device arrives before the timeout period expires the existing thread will be reused. The timeout value is currently hard coded at 5 minutes.

Note that this option can cause one thread (or possibly more) to be created for each device in your configuration. Specifying 0 means there is no limit to the number of threads that can be created.

**-1** Specifiy -1 to cause 'one time only' temporary threads to be created to service each I/O request to a device. Once the I/O request is complete, the thread exits. Subsequent I/Os to the same device will cause another worker thread to be created again.

**1-n** Specifiy a value from 1 to n to set an upper limit to the number of threads that can be created to service any I/O request to any device. Like the "0" option, each thread once finished servicing an I/O request enters an idle state. If a new request arrives before the timeout period expires, the thread is reused. If all threads are busy when a new I/O request arrives a new thread is created only if the specified maximum number of threads have not yet been reached. If the specified maximum number of threads already has been reached then the I/O request is placed in a queue and will be serviced by the first available thread (eg. by whichever thread becomes idle first).

This option was created to address a threading issue, possibly related to the Cygwin phtreads implementation on Windows systems. On Windows systems positive DEVTMAX values are currently not honored and are treated identically as if the value 0 had been specified. The default for non-Windows systems is 0.

### *4.16.4 Examples*

**Example 1:**

Set the upper limit for the number of device threads to 16.

```
DEVTMAX 16
```

**Example 2:**

Allow an unlimited number of device threads.

```
DEVTMAX 0
```

## 4.17 DIAG8CMD (DIAGNOSE 8 command setting)

### *4.17.1 Function*

This parameter specifies whether a command issued through Diagnose 8 will be executed by Hercules as a Hercules commands or not. An optional second argument can be specified to request whether the commands issued using the Diagnose 8 interface will be traced at the console. This may be useful for programs that routinely issue console commands using the Diagnose 8 interface.

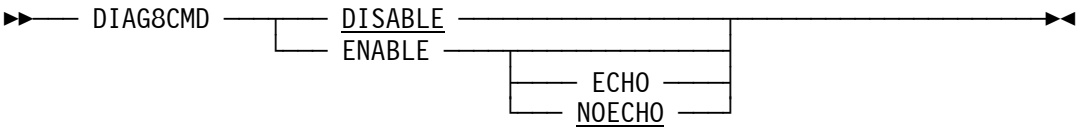
**Caution:** Enabling this feature may have security consequences. When this feature is enabled it is possible for guest operating systems running under Hercules to issue commands directly to the host operating system by means of the Hercules sh (shell) command. This ability may be disabled via the SHCMDOPT statement.

### *4.17.2 Syntax*

**Descriptive**

```
DIAG8CMD {DISABLE | ENABLE [ECHO | NOECHO]}
```

**Diagram**




### *4.17.3 Parameter*

**DISABLE** Commands issued through the Diagnose 8 interface will generate a Specification Exception program interrupt on the issuing CPU. This is the default.

**ENABLED** Commands issued through the Diagnose 8 interface will be executed by Hercules as Hercules commands.

**ECHO** When ECHO is specified, a message is issued as the console is about to issue the command, the command is redisplayed as if it was entered through the console input line, and a final message is issued to indicate the command completed.

**NOECHO** When NOECHO is specified, no such messages are displayed and the command completes silently, except for the output of the command itself if the Diagnose 8 interface did not request a response buffer. This is the default.

The value of ECHO or NOECHO has no effect on command output being placed into a response buffer if the Diagnose 8 interface requested one.

### *4.17.4 Examples*

**Example 1:**

Specify that commands issued through the Diagnose 8 interface are executed as Hercules commands. Additionally issue a message as the console is about to execute the commnd, redisplay the command itself and give a final message, indicating the command has completed.

```
DIAG8CMD ENABLE ECHO
```

## 4.18 ECPSVM (ECPS:VM support status (VM))

### *4.18.1 Function*

This parameter specifies, whether ECPS:VM (Extended Control Program Support : Virtual Machine) support is to be enabled and – if it is enabled – to which level. The purpose of ECPS:VM is to provide to the VM/370 operating system a set of shortcut facilities to perform hypervisor functions (CP Assists) and virtual machine simulation (VM Assists).

Although this feature does not affect VM operating system products operating in XA, ESA or z/Architecture mode, it will affect VM/370 and VM/SP products running under VM/XA, VM/ESA or z/VM. Running VM/370 and VM/SP products under VM/XA, VM/ESA or z/VM should be done with ECPS:VM disabled. ECPS:VM should not be enabled in AP or MP environments. ECPS:VM has no effect on non-VM operating systems. It is however recommended to disable ECPS:VM when running native non-VM operating systems.

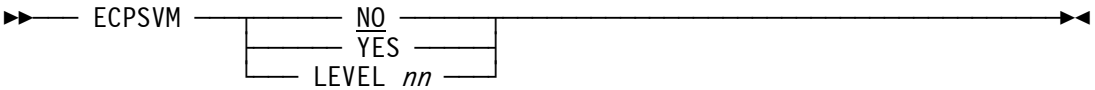
If a specific level is given this value will be reported to the operating system when it issues a Store ECPS:VM level but it does not otherwise alter the ECPS:VM facility operations. Please note that this is a partial implementation.

### *4.18.2 Syntax*

**Descriptive**

```
ECPSVM {NO | YES | LEVEL nn}
```

**Diagram**




### *4.18.3 Parameter*

**NO** ECPS:VM support is disabled. This is the default.

**YES** ECPS:VM support is enabled. The support level that is reported to the operating system is set to 20.

**LEVEL nn** ECPS:VM support is enabled. The value nn specifies the support level that is reported to the operating system.

### *4.18.4 Examples*

**Example 1:**

Enable “Extended Control Program Support : Virtual Machine” support.

```
ECPSVM YES
```

**Example 2:**

Enable “ECPS:VM” support and set the level reported to the operating system to 20.

```
ECPSVM LEVEL 20
```

## 4.19 ENGINES (Processor engine type)

### *4.19.1 Function*

The ENGINES parameter specifies the type of engine for each installed processor. The default engine type is CP. The number of installed processor engines is determined by the MAXCPU configuration statement.

If the ENGINES configuration statement specifies more than MAXCPU engines, the excess engines are ignored. If fewer than MAXCPU engines are specified, the remaining engines are set to type CP (the default).

See the MAXCPU configuration statement for details regarding the MAX_CPU_ENGINES compile time variable.

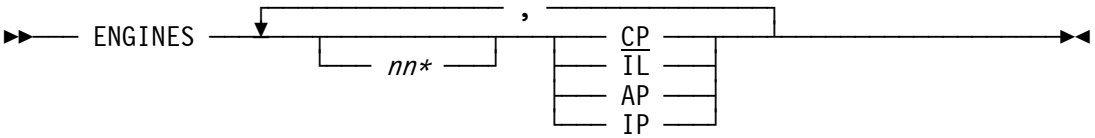
For detailed explanations on the interrelationship between ENGINES, MAXCPU and NUMCPU please see “Appendix B. Configuration of Emulated CPUs”.

### *4.19.2 Syntax*

**Descriptive**

```
ENGINES [nn*] {CP | IL | AP | IP} [, ... ]
```

**Diagram**




### *4.19.3 Parameter*

**nn*** This is an optional repeat count.

**CP** Specifies a processor engine of type CP. This is the default.

**IL** Specifies a processor engine of type IL.

**AP** Specifies a processor engine of type AP.

**IP** Specifies a processor engine of type IP.

### *4.19.4 Examples*

**Example 1:**

Specify 4 engines of type CP, 2 engines of type AP and 2 engines of type IP.

```
ENGINES CP,CP,CP,CP,AP,AP,IP,IP
   or
ENGINES 4*CP,2*AP,2*IP
```

**Example 2:**

Specify 4 engines of type CP, 1 engine of type AP, 1 engine of type IP and 2 engines of type IL.

```
ENGINES CP,CP,CP,CP,AP,IP,IL,IL
   or
ENGINES 4*CP,AP,IP,2*IL
```

**Example 3:**

Specify 3 engines of type CP and 1 engines of type IL.

```
ENGINES CP,CP,CP,IL
   or
ENGINES 3*CP,IL
```

## 4.20 HERCLOGO (Hercules 3270 logo)

### *4.20.1 Function*

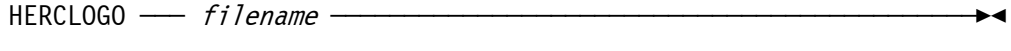
HERCLOGO specifies the logo text file which defines a welcome screen that is presented when a TN3270 terminal connects to a Hercules 3270 device. For details on how to code the Hercules logo file see section 10 (“Hercules 3270 Logo”).

### *4.20.2 Syntax*

**Descriptive**

HERCLOGO *filename*

**Diagram**




### *4.20.3 Parameter*

**filename** The name of a logo text file in the current Hercules directory.

### *4.20.4 Examples*

**Example 1:**

Use the logo text file “HERCLOGO.txt” for the welcome screen.

```
HERCLOGO HERCLOGO.TXT
```

## 4.21 HERCPRIO (Hercules process priority)

### *4.21.1 Function*

The HERCPRIO parameter specifies the process priority for Hercules.

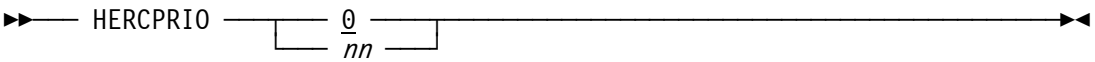
**Caution:** HERCPRIO should not have a higher dispatching priority than the TOD clock and timer thread.

### *4.21.2 Syntax*

**Descriptive**

```
HERCPRIO {0 | nn}
```

**Diagram**




### *4.21.3 Parameter*

**nn** This value specifies the process priority for Hercules. For details on the priority values see section 4.57 (“Process and Thread Priorities”). The default is 0.

### *4.21.4 Examples*

**Example 1:**

Set the Hercules process priority to 0.

```
HERCPRIO 0
```

## 4.22 HTTPPORT (HTTP server port)

### *4.22.1 Function*

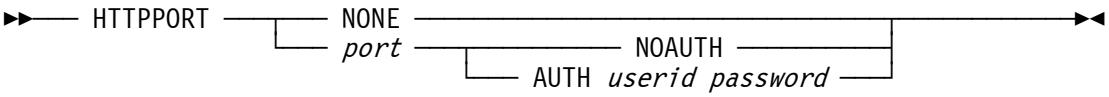
The HTTPPORT statement specifies the port number (in decimal) on which the HTTP server will listen.If no such statement is present, the argument NONE or an invalid port number is given then the HTTP server thread will not be activated.

### *4.22.2 Syntax*

**Descriptive**

```
HTTPPORT {NONE | port {NOAUTH | AUTH userid password}}
```

**Diagram**




### *4.22.3 Parameter*

| | |
|---|---|
| **NONE** | This specifies that the HTTP server is not to be started. |
| **port** | The port number must be either 80 or within the range of 1024 to 65535 inclusive. |
| **NOAUTH** | NOAUTH indicates that no userid and password are required to access the HTTP server. |
| **AUTH** | AUTH indicates that a userid and a password are required to access the HTTP server. The userid and password have to be coded after the AUTH parameter. |
| **userid** | The userid can be any valid string. |
| **password** | The password can be any valid string. |

### *4.22.4 Examples*

**Example 1:**

Set the port number on which the HTTP server will listen to 8081 and specify that authorization is required to access the HTTP server. The userid should be UID0001 and the password should be PSWD0001.

```
HTTPPORT 8081 AUTH UID0001 PSWD0001
```

**Example 2:**

Set the port number on which the HTTP server will listen to 80 and specify that no authorization is required to access the HTTP server.

```
HTTPPORT 80 NOAUTH
```

## 4.23 HTTPROOT (HTTP server root directory)

### *4.23.1 Function*

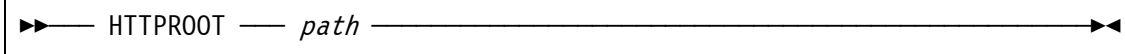
HTTPROOT specifies the full path of the root directory where the HTTP servers files reside. If this parameter is not specified the default value for Win32 builds of Hercules is the directory where the Hercules executables themselves reside. For non–Win32 builds it is the directory specified as the default package installation directory when the Hercules executables were built. This can vary depending on how the Hercules package was built, it is commonly “/usr/local/share/hercules”.

### *4.23.2 Syntax*

**Descriptive**

```
HTTPROOT path
```

**Diagram**




### *4.23.3 Parameter*

**path** The full path of the root directory where the HTTP server’s files reside.

### *4.23.4 Examples*

**Example 1 (for Unix installations):**

Set the root directory for the HTTP servers files under a Unix system to "/usr/local/share/hercules".

```
HTTPROOT /usr/local/share/hercules
```

**Example 2 (for Windows installations):**

Set the root directory for the HTTP servers files under a Windows system to "D:/HERCULES/HTML".

```
HTTPROOT D:/HERCULES/HTML
```

## 4.24 IGNORE (Ignore subsequent INCLUDE errors)

### *4.24.1 Function*

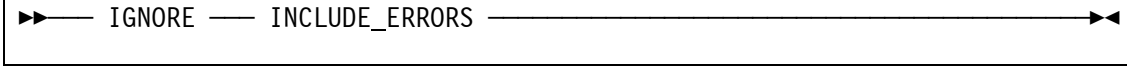
This system parameter indicates that errors caused by subsequent INCLUDE statements - for files which do not exist - should be ignored rather than causing the Hercules startup to be aborted as it would otherwise normally occur

### *4.24.2 Syntax*

**Descriptive**

```
IGNORE INCLUDE_ERRORS
```

**Diagram**




### *4.24.3 Parameter*

**INCLUDE_ERRORS** Indicates that subsequent INCLUDE errors will be ignored.

### *4.24.4 Examples*

**Example 1:**

Specify that errors caused by subsequent INCLUDE statements should be ignored.

```
IGNORE INCLUDE_ERRORS
```

## 4.25 INCLUDE (Include configuration file)

### *4.25.1 Function*

An INCLUDE statement instructs the Hercules initialisation process to treat the contents of the file specified by filepath as if its contents had appeared in the configuration file at the point where the INLCUDE statement appears.

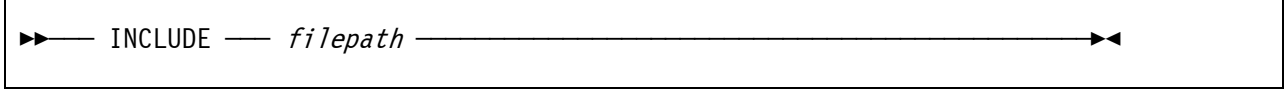
The included file itself may contain yet another INCLUDE statement as long as the maximum nesting depth of 8 is not exceeded.

### *4.25.2 Syntax*

**Descriptive**

```
INCLUDE filepath
```

**Diagram**




### *4.25.3 Parameter*

**filepath** Specifies the path and the filename of the configuration file that is to be included.

### *4.25.4 Examples*

**Example 1:**

Specify that Hercules should include the contents of the file "D:\S390\CONF\CONFIG2.CONF" into the configuration.

```
INCLUDE D:\S390\CONF\CONFIG2.CONF
```

## 4.26 IODELAY (I/O interrupt wait time (LINUX))

### *4.26.1 Function*

IODELAY sets the amount of time in microseconds to wait after an I/O interrupt is ready to be set pending. This value can also be set using the Hercules console. The purpose of this parameter is to circumvent a bug in the Linux/390 and z/Linux dasd.c device driver. The bug is more likely to occur under Hercules than on a real machine as Hercules may present an I/O interrupt sooner than a real machine.

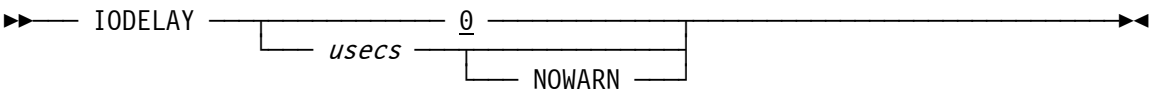
NOTE : OSTAILOR LINUX no longer sets IODELAY to 800 since the problem described above is no longer present in recent versions of the linux kernel.

### *4.26.2 Syntax*

**Descriptive**

```
IODELAY {0 | usecs [NOWARN]}
```

**Diagram**




### *4.26.3 Parameter*

**usecs** Amount of time in microseconds to wait after an I/O interrupt is ready to be set pending.

**NOWARN** If the IODELAY value is non-zero a warning message (HHCCF037W) will be issued unless NOWARN is specified.

### *4.26.4 Examples*

**Example 1:**

Set the amount of time to wait after an I/O interrupt is ready to be set pending to 1000 microseconds.

```
IODELAY 1000
```

**Example 2:**

Set the amount of time to wait after an I/O interrupt is ready to be set pending to 500 microseconds and suppress the warning message HHCCF037W.

```
IODELAY 500 NOWARN
```

## 4.27 LDMOD (Additional dynamic load modules)

### *4.27.1 Function*

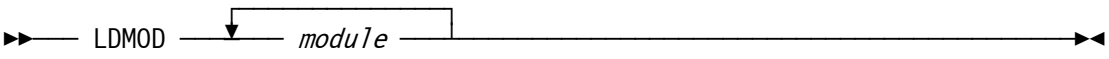
LMOD provides the capability of defining additional modules that will be loaded by the Hercules Dynamic Loader. The default search order is within the Hercules directory and in the default DLL search path. Most systems also support absolute filenames (eg. Names starting with “/” or “.”) in which case the default search path is not taken. Multiple LDMOD statements may be used.

### *4.27.2 Syntax*

**Descriptive**

LMOD *module module module* **...**

**Diagram**




### *4.27.3 Parameter*

**module** A list of modules that are to be loaded by the Hercules Dynamic Loader.

### *4.27.4 Examples*

**Example 1:**

Instruct Hercules to load the additional modules named MODULE01, MODULE02 and MODULE03 from the Hercules directory or the default DLL search path.

```
LDMOD MODULE01 MODULE02 MODULE03
```

## 4.28 LEGACYSENSEID (SENSE ID CCW (x'E4') feature)

### *4.28.1 Function*

This option specifies whether the SENSE ID CCW (X'E4') will be honored for the devices that originally did not support that feature. This includes (but may not be limited to) 3410 and 3420 tape drives, 2311 and 2314 direct access storage devices and 2703 communication controllers. Because those legacy devices didn't originally support this command, and for compatibility reasons, the default is OFF or DISABLE.

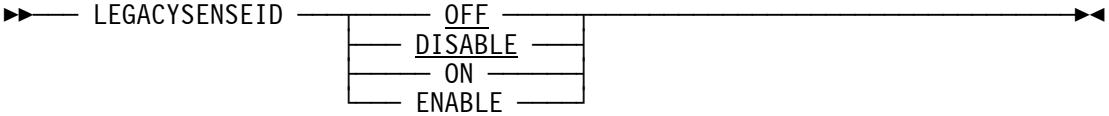
Specify ON or ENABLE if your guest operating system needs the Sense ID support to dynamically detect those devices. Note that most current operating systems will not detect those devices even though Sense ID is enabled because those devices never supported the Sense ID in the first place. This mainly applies to custom built or modified versions of guest operating systems that are aware of this specific Hercules capability.

### *4.28.2 Syntax*

**Descriptive**

```
LEGACYSENSEID [{OFF | DISABLE} | {ON | ENABLE}]
```

**Diagram**




### *4.28.3 Parameter*

**OFF or DISABLE** Specify OFF or DISABLE if your guest operating system does not need the Sense ID support to dynamically detect devices that originally did not support that feature. This is the default.

**ON or ENABLE** Specify ON or ENABLE if your guest operating system needs the Sense ID support to dynamically detect devices that originally did not support that feature.

### *4.28.4 Examples*

**Example 1:**

Specify that the SENSE ID CCW will be honored for devices that originally did not support that feature.

```
LEGACYSENSEID ENABLE
```

## 4.29 LOADPARM (IPL parameter)

### *4.29.1 Function*

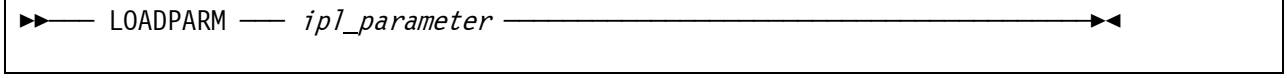
This parameter specifies the eight-character IPL parameter which is used by all MVS based operating systems (MVS 3.8J, MVS/SP, MVS/XA, MVS/ESA, OS/390, z/OS, …) to select the system start parameter.

### *4.29.2 Syntax*

**Descriptive**

LOADPARM *ipl_parameter*

**Diagram**




### *4.29.3 Parameter*

**ipl_parameter** The system parameter used for the IPL of the intended operating system. The parameter is operating system dependent, consult the relevant operating system documentation for details.

### *4.29.4 Examples*

**Example 1:**

Use 0A8200M1 as system parameter for the IPL of the operating system.

```
LOADPARM 0A8200M1
```

## 4.30 LOGOPT (Log options)

### *4.30.1 Function*

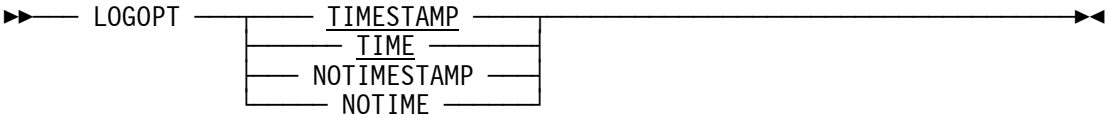
The LOGOPT statement specifies the Hercules log options. It allows to insert or suppress the time stamp in front of each log message

### *4.30.2 Syntax*

**Descriptive**

```
LOGOPT {TIMESTAMP | TIME | NOTIMESTAMP | NOTIME}
```

**Diagram**




### *4.30.3 Parameter*

| | |
|---|---|
| **TIMESTAMP** | Insert a time stamp in front of each log message. This is the same as using TIME. |
| **NOTIMESTAMP** | Display log messages without a timestamp. This is the same as using NOTIME. |
| **TIME** | Insert a time stamp in front of each log message. This is the same as using TIMESTAMP. |
| **NOTIME** | Display log messages without a timestamp. This is the same as using NOTIMESTAMP. |

### *4.30.4 Examples*

**Example 1:**

Specify that messages in the log are to be time stamped.

```
LOGOPT TIMESTAMP
```

## 4.31 LPARNAME (LPAR name returned by DIAG x'204')

### *4.31.1 Function*

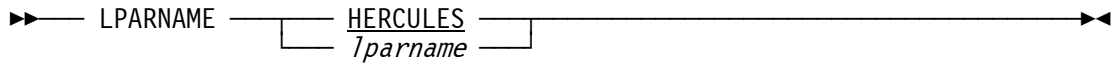
LPARNAME defines the LPAR name returned by DIAG x'204'.

### *4.31.2 Syntax*

**Descriptive**

LPARNAME {HERCULES | *lparname*}

**Diagram**




### *4.31.3 Parameter*

**lparname** Maximum 8 byte character name for the LPAR. The default is HERCULES.

### *4.31.4 Examples*

**Example 1:**

Specify HERCULES as LPAR name returned by DIAG x'204'.

```
LPARNAME  HERCULES
```

## 4.32 LPARNUM (LPAR identification number)

### *4.32.1 Function*

The LPARNUM configuration statement sets the LPAR identification number. It can be one or two hexa-decimal digits. If one digit is specified then STIDP stores a format-0 CPU ID containing the logical CPU number, the LPAR number, and the last four digits of the CPU serial number. If two digits are specified then STIDP stores a format-1 CPU ID containing the two-digit LPAR identifier followed by the last four digits of the CPU serial number.

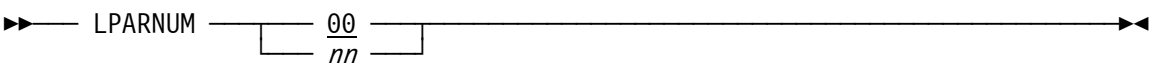
If LPARNUM is not specified in the configuration file then STIDP returns a CPUID as before, where the leftmost digit of the serial number is replaced by the logical CPU number if the serial number begins with a zero.

### *4.32.2 Syntax*

**Descriptive**

```
LPARNUM {00 | nn}
```

**Diagram**




### *4.32.3 Parameter*

**xx** Specifies the one- or two-digit hexadecimal LPAR identification number stored by the STIDP instruction. If a one-digit number is specified then STIDP stores a format-0 CPU ID. If a two-digit number is specified then STIDP stores a format-1 CPU-ID.

### *4.32.4 Examples*

**Example 1:**

Set the LPAR identification number to x'21' (Format-1 CPU-ID).

```
LPARNUM 21
```

**Example 2:**

Set the LPAR identification number to x'A' (Format-0 CPU-ID).

```
LPARNUM A
```

## 4.33 MAINSIZE (Main storage in MB)

### *4.33.1 Function*

Specifies the size of the main storage in megabytes. The actual upper limit of the main storage is determined by the host system's architecture, operating system, and on some systems the amount of physical memory and paging space you have available. The value must not exceed 4095 (almost 4GB) for 32-bit platforms and 4,294,967,295 (about 4 million GB) for 64-bit platforms.

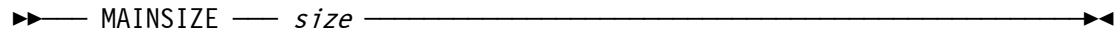
The practical limit depends on the maximum amount of storage that can be obtained by malloc (usually around 1 GB on 32-bit platforms; on 64-bit platforms the value should only be limited by available paging space).

### *4.33.2 Syntax*

**Descriptive**

```
MAINSIZE size
```

**Diagram**




### *4.33.3 Parameter*

**size** A valid decimal number whose value is at least 2. The upper limit must not exceed 4095 (almost 4GB) for 32-bit platforms and 4,294,967,295 (about 4 million GB) for 64-bit platforms.

### *4.33.4 Examples*

**Example 1:**

Set the size of the main storage to 64 MB.

```
MAINSIZE 64
```

**Example 1:**

Set the size of the main storage to 4 GB.

```
MAINSIZE 4096
```

## 4.34 MANUFACTURER (Manufacturer name returned by STSI instruction)

### *4.34.1 Function*

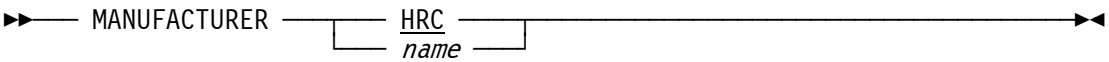
MANUFACTURER specifies the manufacturer name returned by the STSI instruction. The default name is HRC.

### *4.34.2 Syntax*

**Descriptive**

```
MANUFACTURER {HRC | name}
```

**Diagram**




### *4.34.3 Parameter*

**name** Any name with a maximum length of four characters. If MANUFACTURER is not specified in the configuration file then the default name is HRC.

### *4.34.4 Examples*

**Example 1:**

Set the manufacturer name returned by the STSI instruction to “HERC”.

```
MANUFACTURER HERC
```

## 4.35 MAXCPU (Maximum number of CPUs)

### *4.35.1 Function*

The MAXCPU configuration statement specifies the maximum number of installed processor engines. The NUMCPU statement specifies the number of engines which will be configured online at startup time. All processors are CP engines unless otherwise specified by the ENGINES statement.

MAX_CPU_ENGINES is a compile-time variable which sets an upper limit on the value of MAXCPU. The value of MAX_CPU_ENGINES is displayed in the build information message on the Hercules control panel at startup time. To change the value of MAX_CPU_ENGINES you must rebuild Hercules. For Unix builds, specify "./configure --enable-multi-cpu=*nn*" before performing make. For Windows builds, specify "SET MAX_CPU_ENGINES=*nn*" before performing nmake.

MAX_CPU_ENGINES may be up to 128 on 64-bit Linux platforms. On Windows, and on all 32-bit platforms, the maximum value is 64. For performance reasons, values above 32 are not recommended for 32-bit platforms. If MAX_CPU_ENGINES is set to 1 then multiprocessing is disabled. See also the NUMCPU statement for a discussion of the performance implications of MAX_CPU_ENGINES.

The value of MAXCPU can not exceed the value of MAX_CPU_ENGINES. If MAXCPU is not specified then the default value is the value of MAX_CPU_ENGINES.

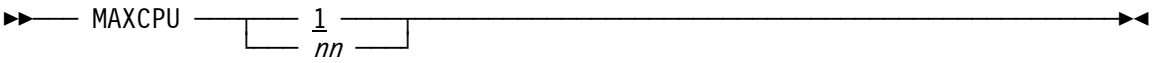
For detailed explanations on the interrelationship between MAXCPU, ENGINES and NUMCPU please see “Appendix B. Configuration of Emulated CPUs”.

### *4.35.2 Syntax*

**Descriptive**

```
MAXCPU {1 | nn}
```

**Diagram**




### *4.35.3 Parameter*

**nn** Specifies the maximum number of installed processor engines. The value of MAXCPU can not exceed the value of MAX_CPU_ENGINES. If MAXCPU is not specified in the configuration file then the default value is the value of MAX_CPU_ENGINES.

### *4.35.4 Examples*

**Example 1:**

Set the maximum number of installed processor engines to 8.

MAXCPU 8

## 4.36 MODEL (Model names returned by STSI instruction)

### *4.36.1 Function*

MODEL specifies the model names returned by the STSI instruction. If two operands are supplied, the first is the hardware model name (CPC ND model) and the second is the capacity model name (CPC SI model).

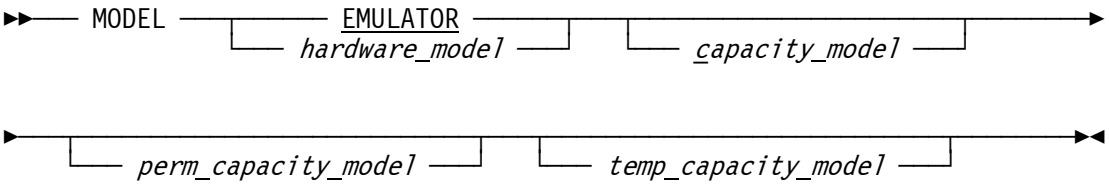
If only one operand is supplied, it is used as both the hardware model name and the capacity model name. The optional third and fourth operands specify the permanent capacity model name and the temporary capacity model name respectively. The default model name is EMULATOR.

### *4.36.2 Syntax*

**Descriptive**

```
MODEL {EMULATOR | hardware_model} [capacity_model]
      [perm_capacity_model] [temp_capacity_model]
```

**Diagram**




### *4.36.3 Parameter*

**hardware_model** This specifies the hardware model name. This can be any name with a maximum length of 16 characters. If the MODEL system parameter is not specified in the configuration file then the default name is EMULATOR.

**capacity_model** This specifies the capacity model name. If this parameter is omitted the hardware_model name is used as capacity_model name.

**perm_capacity_model** This optional parameter specifies the permanent capacity model name.

**temp_capacity_model** This optional parameter specifies the temporary capacity model name.

### *4.36.4 Examples*

**Example 1:**

Set the model name returned by the STSI instruction to “EMULATOR”.

```
MODEL EMULATOR
```

## 4.37 MODPATH (Dynamic load module path)

### *4.37.1 Function*

MODPATH specifies the path where dynamic load modules are loaded from. If a MODPATH statement is coded then the path on the MODPATH statement is searched before the default path is searched. When a relative path is specified it is interpreted as a relative path within the default search path. If an absolute path is coded it is interpreted as such.

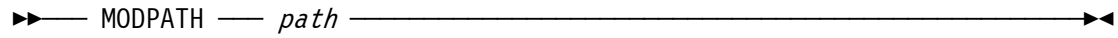
The default MODPATH is “hercules” which means modules are loaded from the directory hercules within the default LD_LIBRARY_PATH.

### *4.37.2 Syntax*

**Descriptive**

```
MODPATH path
```

**Diagram**




### *4.37.3 Parameter*

**path** Path where dynamic load modules are loaded from.

### *4.37.4 Examples*

**Example 1:**

Set the path from where dynamic load modules are loaded to “D:/HERCULES/MODIFICATIONS”.

```
MODPATH D:/HERCULES/MODIFICATIONS
```

## 4.38 MOUNTED_TAPE_REINIT (Control tape initialization)

### *4.38.1 Function*

This system parameter specifies whether reinitialization of tape drive devices via the DEVINIT command in order to mount a new tape should be allowed if there is already a tape mounted on the drive. This option is meant as a safety mechanism to protect against accidentally unmounting a tape from the wrong device as a result of a simple typing error and thereby eventually cancelling a potentially important tape job. Please note that for SCSI tape drives the "DEVINIT nnnn *" command has no effect. The tape must be unmounted manually since it is a real physical device and not emulated via a disk file like '.AWS' or '.HET' tapes.

Specifiying ALLOW, the default, indicates that new tapes may be mounted via "DEVINIT nnnn new-tape-filename" irrespective of whether or not there is already a tape mounted on the drive.

Specifying DISALLOW prevents new tapes from being mounted if one is already mounted. Before the new tape can be mounted the existing one has first to be unmounted (via the "DEVINIT nnnn *" command). Otherwise the DEVINIT attempt to mount the new tape is rejected.

### *4.38.2 Syntax*

**Descriptive**

```
MOUNTED_TAPE_REINIT {ALLOW | DISALLOW}
```

**Diagram**

### *4.38.3 Parameter*

**ALLOW** Indicates that new tapes may be mounted irrespective of wether or not there is already a tape mounted on the drive. This is the default.

**DISALLOW** Prevents new tapes from being mounted if one is already mounted on the drive. Before the new tape can be mounted the currently mounted tape must first to be unmounted.

### *4.38.4 Examples*

**Example 1:**

Specify that new tapes are not being mounted if one is already mounted on the drive.

```
MOUNTED_TAPE_REINIT DISALLOW
```

## 4.39 NUMCPU (Number of emulated CPUs)

### *4.39.1 Function*

NUMCPU specifies the number of emulated processor engines which will be configured online at startup time. All processors are CP engines unless otherwise specified by the ENGINES configuration statement. NUMCPU can not exceed the value of the MAXCPU configuration statement. If NUMCPU is less than MAXCPU then the remaining engines can be configured online later.

Multiprocessor emulation works best if your host system actually has more than one physical CPU, but you can still emulate multiple CPUs nevertheless even on a uniprocessor system (and you might even achieve a small performance benefit when you do). There is little point, however, in specifying NUMCPU greater than 1 unless your guest operating system (running under Hercules) is actually able to support multiple CPUs.

If you do not actually need multiprocessor emulation, then setting MAX_CPU_ENGINES to 1 at compile time might even produce a slight performance advantage too.

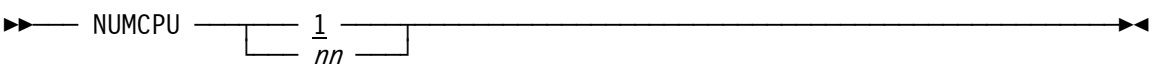
For detailed explanations on the interrelationship between NUMCPU, MAXCPU and ENGINES please see “Appendix B. Configuration of Emulated CPUs”.

### *4.39.2 Syntax*

**Descriptive**

```
NUMCPU {1 | nn}
```

**Diagram**




### *4.39.3 Parameter*

**number** The number of emulated CPUs. NUMCPU can not exceed the value of the MAXCPU configuration statement. If NUMCPU is less than MAXCPU then the remaining engines can be configured online later.

### *4.39.4 Examples*

**Example 1:**

Set the number of emulated CPUs to 4.

```
NUMCPU 4
```

## 4.40 NUMVEC (Number of vector facilities)

### *4.40.1 Function*

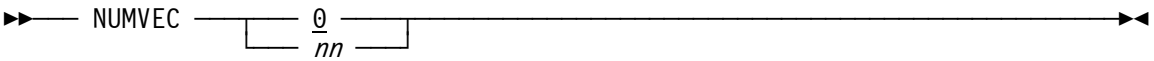
NUMVEC defines the number of emulated vector facilities. Default is one per CPU (see also the NUMCPU parameter). The vector facility is only available in ESA/390 mode by default.

### *4.40.2 Syntax*

**Descriptive**

```
NUMVEC {0 | nn}
```

**Diagram**




### *4.40.3 Parameter*

**number** The number of desired vector facilities.

### *4.40.4 Examples*

**Example 1:**

Set the number of emulated vector facilities to 1.

```
NUMVEC 1
```

## 4.41 OSTAILOR (Intended operating system)

### *4.41.1 Function*

OSTAILOR lets you specify the intended operating system. The effect of this parameter is to reduce control panel message traffic by selectively suppressing trace messages for program checks which are considered normal in the specified environment. If this statement is omitted exception messages for program checks 10, 11, 16, and 1C are suppressed.

The argument *QUIET* suppresses all exception messages whereas the argument *NULL* suppresses none of them. The other options do suppress some messages and do not suppress other messages depending on the specified operating system.

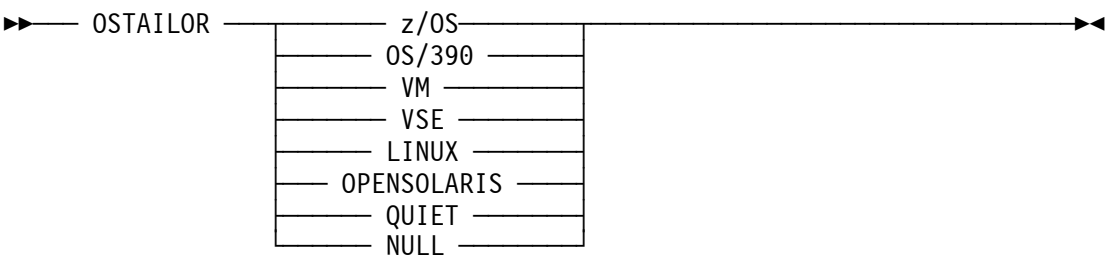
See also the PGMTRACE command which allows to further fine tune the tracing of program interrupt exceptions.

### *4.41.2 Syntax*

**Descriptive**

```
OSTAILOR {z/OS | OS/390 | VM | VSE | LINUX | OPENSOLARIS | QUIET | NULL}
```

**Diagram**




### *4.41.3 Parameter*

| | |
|---|---|
| **z/OS** | Code z/OS if you intend to run z/OS. |
| **OS/390** | Code OS/390 if you intend to run MVS/370, MVS/XA, MVS/ESA, OS/390. |
| **VM** | Code VM if you intend to run VM/370, VM/ESA or z/VM. |
| **VSE** | Code VSE if you intend to run VSE/370, VSE/ESA or z/VSE. |
| **LINUX** | Code Linux you intend to run Linux/390 or Linux for z/Series. |
| **OpenSolaris** | Code OpenSolaris you intend to run OpenSolaris for z/Series. |
| **QUIET** | QUIET discards all exception messages. |
| **NULL** | NULL allows all exception messages to be logged. |

### *4.41.4 Examples*

**Example 1:**

Specify Linux as the intended operating system and selectively suppressing trace messages for program checks which are considered normal in the Linux environment.

```
OSTAILOR LINUX
```

**Example 2:**

Do not specify the intended operating system and allow all exception messages to be logged.

```
OSTAILOR NULL
```

## 4.42 PANRATE (Panel refresh rate)

### *4.42.1 Function*

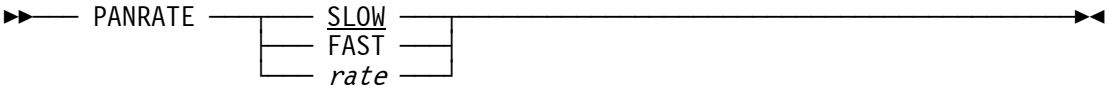
PANRATE defines the rate, in milliseconds, at which the Hercules hardware console (HMC) will be refreshed. Please be aware that the panel refresh rate can reduce overall Hercules performance if PANRATE is set to fast.

### *4.42.2 Syntax*

**Descriptive**

```
PANRATE {SLOW | FAST | rate}
```

**Diagram**




### *4.42.3 Parameter*

**SLOW** SLOW is a synonym for a Hercules hardware console refresh rate of 500 milliseconds. This is also the default, if PANRATE is not specified.

**FAST** FAST is a synonym for a Hercules hardware console refresh rate of 50 milliseconds.

**rate** Any value between 1 (10) and 5000 milliseconds. A value less than the Linux system clock tick interval (10 on Intel platforms, 1 on Alpha platforms) or a value of more than 5000 will be rejected.

### *4.42.4 Examples*

**Example 1:**

Set the panel refresh rate to 1 second.

```
PANRATE 1000
```

**Example 2:**

Set the panel refresh rate to 50 milliseconds.

```
PANRATE FAST
```

## 4.43 PANTITLE (Hercules console window title)

### *4.43.1 Function*

PANTITLE specifies an optional console window title-bar string to be used in place of the default supplied by the windowing system. This option allows one to distinguish between different Hercules sessions when running more than one instance of Hercules on the same machine.

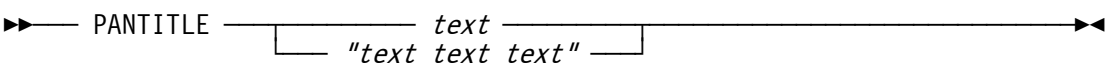
The option takes effect only when the Hercules console is displayed on an xterm terminal (commonly used on Unix systems) or in a Windows command prompt window. Note that this option has no effect when Hercules is run under the control of the Hercules Windows GUI since Hercules's console window is hidden in favour of using the GUI's window instead.

### *4.43.2 Syntax*

**Descriptive**

```
PANTITLE {text | "text text text"}
```

**Diagram**




### *4.43.3 Parameter*

**text** Specifies the optional console window title-bar string to be used. If the value contains any blanks it must be enclosed within double-quotes (").

### *4.43.4 Examples*

**Example 1:**

Set the console window title-bar string to “Hercules Emulator HMC”.

```
PANTITLE "Hercules Emulator HMC"
```

**Example 2:**

Set the console window title-bar string to “Hercules”.

```
PANTITLE Hercules
```

## 4.44 PGMPRDOS (LPP license setting)

### *4.44.1 Function*

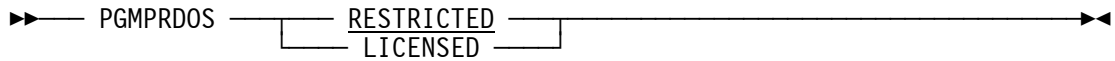
PGMPRDOS specifies whether or not Hercules will run licensed program product (LPP) ESA or z/Architecture operating systems.

### *4.44.2 Syntax*

**Descriptive**

```
PGMPRDOS {RESTRICTED | LICENSED}
```

**Diagram**




### *4.44.3 Parameter*

**RESTRICTED** When PGMPRDOS is set to RESTRICTED, Hercules will stop all CPUs when a licensed program product operating systems is detected. RESTRICTED is the default.

**LICENSED** Setting PGMPRDOS to LICENSED will allow you to run licensed program product operating systems normally. This parameter has no effect on Linux/390, Linux for z/Series, or any 370-mode operating system. If you are running Hercules under the Windows GUI a pop up window appears during startup which must be acknowledged before the startup continuous.

### *4.44.4 Examples*

**Example 1:**

Allow licensed program product operating systems to run normally.

```
PGMPRDOS LICENSED
```

**Example 2:**

Disallow licensed program product operating systems to run.

```
PGMPRDOS RESTRICTED
```

## 4.45 PLANT (Plant name returned by STSI instruction)

### *4.45.1 Function*

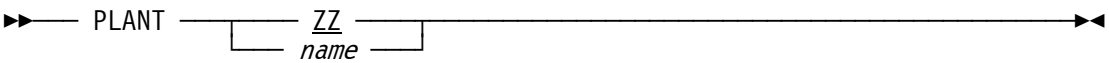
PLANT specifies the plant name returned by the STSI instruction. The default plant name is ZZ.

### *4.45.2 Syntax*

**Descriptive**

PLANT {ZZ | *name*}

**Diagram**




### *4.45.3 Parameter*

**name** Any name with a maximum length of four characters. If the PLANT system parameter is not specified in the configuration file then the default name is ZZ.

### *4.45.4 Examples*

**Example 1:**

Specify “HERC” as the plant name returned by the STSI instruction.

```
PLANT HERC
```

## 4.46 SCLPROOT (SCLP base directory)

### *4.46.1 Function*

The SCLPROOT configuration statement sets the SCLP base directory. If a directory is given then SCLP disk I/O for the specified directory path is enabled. NONE disables SCLP disk I/O.

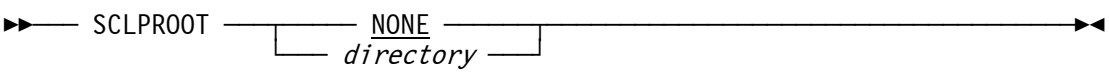
A subsequent list-directed IPL resets the path to the location of the .ins file, and a subsequent CCW-type IPL disables SCLP disk I/O.

### *4.46.2 Syntax*

**Descriptive**

```
SCLPROOT {NONE | directory}
```

**Diagram**




### *4.46.3 Parameter*

**NONE** Disables SCLP disk I/O.

**directory** Specifies the directory from which SCLP disk I/O is allowed. A subsequent IPL of an .ins file or a subsequent CCW-type IPL will override this.

### *4.46.4 Examples*

**Example 1:**

Disable SCLP disk I/O.

```
SCLPROOT NONE
```

**Example 2:**

Specifies “D:\SCLP\DISK” as the directory from which SCLP disk I/O is allowed.

```
SCLPROOT   D:\SCLP\DISK
```

## 4.47 SHCMDOPT (Shell command option)

### *4.47.1 Function*

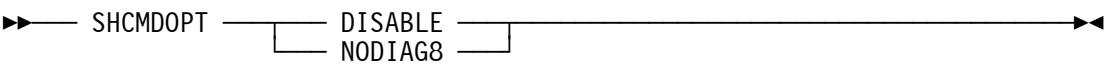
This statement defines the behaviour of the shell (sh) command.

### *4.47.2 Syntax*

**Descriptive**

```
SHCMDOPT {DISABLE | NODIAG8}
```

**Diagram**




### *4.47.3 Parameter*

**DISABLE** When set to `DISABLE`, `sh` (shell) commands are globally disabled and will result in an error if entered either directly via the Hercules hardware console or programmatically via the DIAG8CMD interface.

**NODIAG8** When set to `NODIAG8` only the programmatic execution of shell commands via the the diagnose 8 interface is disabled; `sh` (shell) commands entered directly via the Hercules hardware console will still work.

**NOTE:** *"entered directly via the Hercules hardware console"* includes commands entered via the HTTP server facility or entered via .rc "run command" scripts.

### *4.47.4 Examples*

**Example 1:**

Disable the programmatic execution of shell commands via the the diagnose 8 interface but allow shell commands entered directly via the Hercules hardware console.

```
SHCMDOPT NODIAG8
```

**Example 2:**

Globally disable execution of shell commands.

```
SHCMDOPT DISABLE
```

## 4.48 SHRDPORT (Shared device server port)

### *4.48.1 Function*

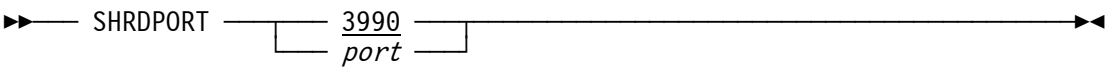
SHRDPORT defines the port number (in decimal) on which the Shared Device Server will listen. The Shared Device Server will allow other Hercules instances to access devices on this instance. Currently only DASD devices may be shared. The default port is 3990. If you decide to use a different port number then you must specify this port number on the device statements for the other Hercules instances. If no SHRDPORT statement is present then the Shared Device Server thread will not be activated.

### *4.48.2 Syntax*

**Descriptive**

```
SHRDPORT {3990 | port}
```

**Diagram**




### *4.48.3 Parameter*

**3990** This is the default port for the Shared Device Server. If this port number is used then the port number is not required to be coded on the device statements for other Hercules instances.

**port** Any valid port number.

### *4.48.4 Examples*

**Example 1:**

Specify 3990 as the port used for the Shared Device Server.

```
SHRDPORT 3990
```

## 4.49 SYSEPOCH (Base date for TOD clock)

### *4.49.1 Function*

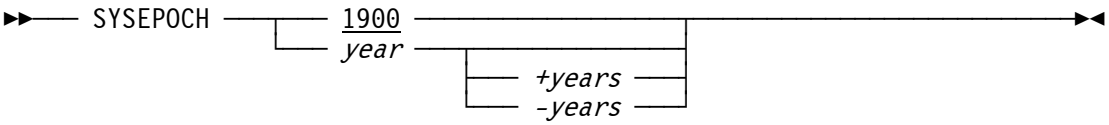
Specifies the base date for the TOD clock. Use the default value (1900) for all systems except OS/360. Use 1960 for OS/360. Values other than these were formerly used to offset the TOD clock by a number of years to move the date before the year 2000 for non-Y2K-compliant operating systems. This use is deprecated and support will be removed in a future release after which only values of 1900 or 1960 will be accepted.

### *4.49.2 Syntax*

**Descriptive**

```
SYSEPOCH {1900 | 1960 | year [+years | -years]}
```

**Diagram**




### *4.49.3 Parameter*

**year** This is the base date for the TOD clock. The only supported values for SYSEPOCH are currently 1900 and 1960. Any other value will produce a warning message showing the equivalent values to specify in the SYSEPOCH statement. 1900 is the default.

**+/- years** An optional year offset may be specified and will be treated as though it had been specified using the YROFFSET statement.

### *4.49.4 Examples*

**Example 1:**

Specify year 1900 as the base date for the TOD clock.

```
SYSEPOCH 1900
```

## 4.50 TIMERINT (Internal timer update interval)

### *4.50.1 Function*

The TIMERINT parameter sets the internal timer update interval in microseconds. This parameter specifies how frequently Hercules's internal timers-update thread updates the TOD clock, CPU Timer and other architectural related clock/timer values.

The default interval is 50 microseconds which strikes a reasonable balance between clock accuracy and overall host performance. The minimum allowed value is 1 microsecond and the maximum is 1,000,000 microseconds (one second).

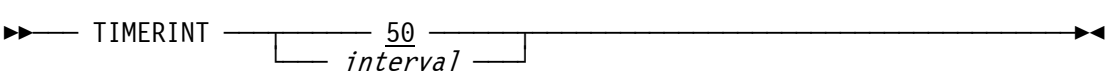
***CAUTION:*** *While a lower TIMERINT value may help increase the accuracy of the guest's TOD clock and CPU Timer values it could also have severe negative impact on the overall performance of the host operating system. This is especially true when a low TIMERINT value is coupled with a high HERCPRIO and TODPRIO priority setting. Exercise extreme caution when choosing your desired TIMERINT in relationship to your chosen HERCPRIO and TODPRIO priority settings.*

### *4.50.2 Syntax*

**Descriptive**

```
TIMERINT {50 | interval}
```

**Diagram**




### *4.50.3 Parameter*

**interval** Specifies the timer update interval in microseconds. The minimum allowed value for the interval is 1 microsecond and the maximum is 1,000,000 microseconds (one second).

### *4.50.4 Examples*

**Example 1:**

Set the timer update interval to 100 microseconds.

```
TIMERINT 100
```

## 4.51 TODDRAG (TOD clock drag factor)

### *4.51.1 Function*

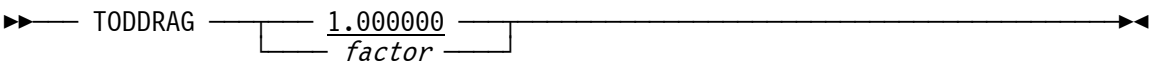
TODDRAG specifies the TOD clock drag factor. This parameter can be used to slow down or speed up the TOD clock by a factor of *nn*. A significant slowdown can improve the performance of some operating systems which consume significant amounts of CPU time processing timer interrupts. A drag factor of 2.0 slows down the clock by 50%, a drag factor of 0.5 doubles the speed of the clock, a drag factor of 1.01 slows down the clock by 1% and 0.99 speeds up the clock by 1%.

### *4.51.2 Syntax*

**Descriptive**

```
TODDRAG {1.000000 | factor}
```

**Diagram**




### *4.51.3 Parameter*

**factor** The factor by which the TOD clock will be slowed down or sped up. The default factor is 1.000000.

### *4.51.4 Examples*

**Example 1:**

Slow down the TOD clock by 50%.

```
TODDRAG 2
```

**Example 2:**

Double the speed of the TOD clock.

```
TODDRAG 0.5
```

## 4.52 TODPRIO (Timer thread process priority)

### *4.52.1 Function*

With this parameter you can specify the priority of the TOD clock and the timer threads.

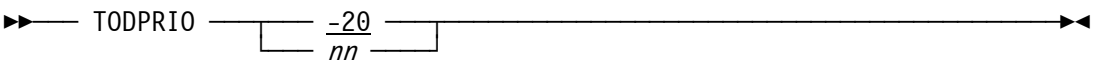
**Caution:** TODPRIO should be given a dispatching priority equal to or higher than any other thread (CPUPRIO, DEVPRIO, HERCPRIO) within Hercules.

### *4.52.2 Syntax*

**Descriptive**

```
TODPRIO {-20 | nn}
```

**Diagram**




### *4.52.3 Parameter*

**nn** This value specifies the priority of the TOD clock and the timer thread. For details on the priority values see section 4.57 (“Process and Thread Priorities”). The default for TODPRIO is -20.

### *4.52.4 Examples*

**Example 1:**

Set the priority of the TOD clock and the timer threads to -20.

```
TODPRIO -20
```

## 4.53 TRACEOPT (Instruction trace display option)

### *4.53.1 Function*

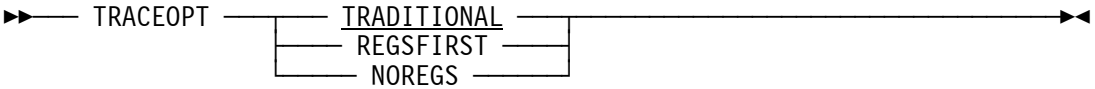
TRACEOPT sets the Hercules instruction tracing display option. In addition to the TRACEOPT configuration file statement there is also a corresponding TRACEOPT console command to dynamically display and/or update the current setting at any time.

### *4.53.2 Syntax*

**Descriptive**

```
TRACEOPT {TRADITIONAL | REGSFIRST | NOREGS}
```

**Diagram**




### *4.53.3 Parameter*

**TRADITIONAL** TRADITIONAL (the default), displays the registers following the instruction about to be executed such that pressing enter (to execute the displayed instruction) then shows the next instruction to be executed followed by the updated registers display.

**REGSFIRST** REGSFIRST displays the current register contents followed by the instruction about to be executed such that pressing enter (to execute the displayed instruction) then shows the updated registers followed by the next instruction to be executed.

**NOREGS** NOREGS suppresses the registers display altogether and shows just the instruction to be executed.

### *4.53.4 Examples*

**Example 1:**

Set the tracing display option to display the registers following the instruction about to be executed.

```
TRACEOPT TRADITIONAL
```

## 4.54 TZOFFSET (TOD clock offset from GMT)

### *4.54.1 Function*

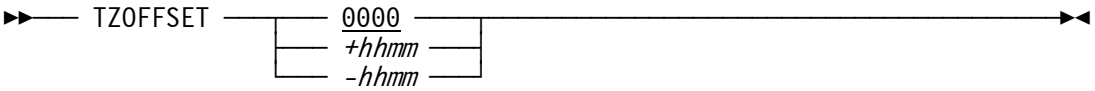
TZOFFSET is used to define the offset of the TOD clock from the current system time. For GMT use the default value (0000). For timezones west of Greenwich specifiy a negative value (example: -0500 for US Eastern Standard Time, -0800 for US Pacific Standard Time). For timezones east of Greenwich, specifiy a positive value (example: +0100 for Central European Time, +0930 for South Australian Time).

### *4.54.2 Syntax*

**Descriptive**

```
TZOFFSET {0000 | +hhmm | -hhmm}
```

**Diagram**




### *4.54.3 Parameter*

| | |
|---|---|
| **0000** | GMT time (0000 is the default value). Please note that this is also the correct setting if your system time (the time of the operating system on which Hercules is running) is set to local time rather than GMT. |
| **+hhmm** | Use a positive time in hours and minutes for timezones east of Greenwich. |
| **-hhmm** | Use a negative time in hours and minutes for timezones west of Greenwich. |

### *4.54.4 Examples*

**Example 1:**

Set the offset of the TOD clock from the current system time to Central European Time.

```
TZOFFSET +0100
```

## 4.55 XPNDSIZE (Expanded storage in MB)

### *4.55.1 Function*

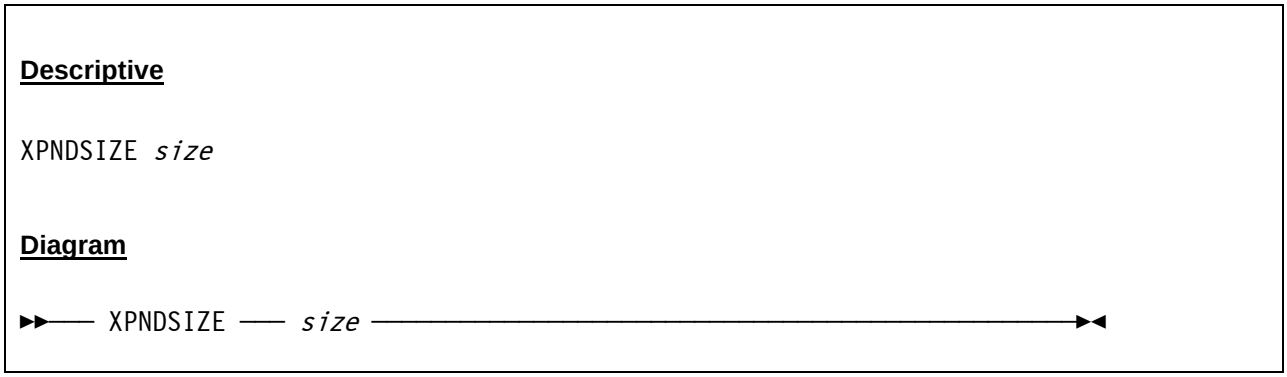
Specifies the expanded storage size in megabytes.

### *4.55.2 Syntax*

**Descriptive**

```
XPNDSIZE size
```

**Diagram**




### *4.55.3 Parameter*

**nnnnnnn** A decimal number in the range of 0 to 16777215. The actual upper limit is determined by your host system's architecture and operating system and on some systems the amount of physical memory and paging space you have available.

### *4.55.4 Examples*

**Example 1:**

Set the size of the expanded storage to 256 MB.

```
XPNDSIZE 256
```

**Example 2:**

Set the size of the expanded storage to 2 GB.

```
XPNDSIZE 204
```

## 4.56 YROFFSET (TOD clock offset from actual date)

### *4.56.1 Function*

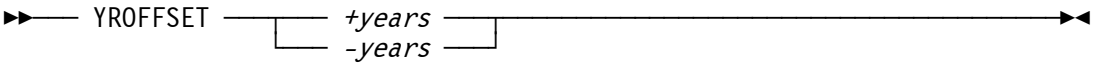
Specifies a number of years to offset the TOD clock from the actual date. Positive numbers will move the clock forward in time while negative numbers will move it backward. A common value for non-Y2K-compliant operating systems is YROFFSET -28 which has the advantage that the day of the week and the presence or absence of February 29 is the same as the current year.

### *4.56.2 Syntax*

**Descriptive**

YROFFSET {*+years* | *-years*}

**Diagram**




### *4.56.3 Parameter*

**+/-years** Years specifies a number of years to offset the TOD clock from the actual date. This value may not be specified as greater than ±142 years, the total range of the TOD clock. Specifying a value that causes the computed TOD clock year to be earlier than the value of SYSEPOCH or more than 142 years later than SYSEPOCH will produce unexpected results.

### *4.56.4 Examples*

**Example 1:**

Specify 28 years to offset the TOD clock from the actual date.

```
YROFFSET    -28
```

## 4.57 Process and Thread Priorities

This section covers details regarding the priority settings within Hercules. The relevant system parameters are:

- CPUPRIO
- DEVPRIO
- HERCPRIO
- TODPRIO

### *4.57.1 Process Priorities*

Under Linux a process is a thread and thread priority information applies instead.

For Windows the following conversions are used for translating Unix process priorities to Windows priority classes:

| **Unix Process Priority** | **Windows Priority Class** | **Meaning** |
|---|---|---|
| -20 to -16 | Realtime | Process that has the highest possible priority. The threads of the process preempt the threads of all other processes, including operating system processes performing important tasks. For example, a real-time process that executes for more than a very brief interval can cause disk caches not to flush or cause the mouse to be unresponsive. |
| -15 to -9 | High | Process that performs time-critical tasks that must be executed immediately. The threads of the process preempt the threads of normal or idle priority class processes. An example is the Task List, which must respond quickly when called by the user, regardless of the load on the operating system. Use extreme care when using the high-priority class, because a high-priority class application can use nearly all available CPU time. |
| -8 to -1 | Above Normal | Process that has priority above the Normal class but below the High class. |
| 0 to 7 | Normal | Process with no special scheduling needs. |
| 8 to 15 | Below Normal | Process that has priority above the Idle class but below the Normal class. |
| 16 to 20 | Low | Process whose threads run only when the system is idle. The threads of the process are preempted by the threads of any process running in a higher priority class. An example is a screen saver. The idle-priority class is inherited by child processes. |

**Table 4: Process Priority Conversions**

**Caution:** On Windows, the value you choose for your Process Priority has a direct impact on how your Thread Priorities are interpreted! You should never modify one without understanding what impact your doing so might have on the other!

### *4.57.2 Thread Priorities*

On Linux/Unix hosts Hercules needs to be a setuid root program to allow it to reset its dispatching priority to a high (negative) value (i.e. “chown root.root hercules; chmod +s hercules”).

For Windows the following conversions are used for translating Linux/Unix thread priorities to Windows thread priorities:

| **Unix Thread Priority** | **Windows Thread Priority** | **Meaning** |
|---|---|---|
| -20 to -16 | Time Critical | Base priority of 15 for Idle, Below Normal, Normal, Above Normal, or High class processes, and a base priority of 31 for Realtime class processes. |
| -15 to -9 | Highest | Priority 2 points above the priority class. |
| -8 to -1 | Above Normal | Priority 1 point above the priority class. |
| 0 to 7 | Normal | Normal priority for the priority class. |
| 8 to 15 | Below Normal | Priority 1 point below the priority class. |
| 16 to 19 | Lowest | Priority 2 points below the priority class. |
| 20 | Idle | Base priority of 1 for Idle, Below Normal, Normal, Above Normal, or High class processes, and a base priority of 16 for Realtime class processes. |

**Table 5: Thread Priority Conversions**

**Caution:** On Windows, your thread priority is interpreted differently based on your chosen process priority setting! You should never modify your thread priority settings without first reviewing your chosen process priority setting!

# 5. Device Definition Descriptions

## 5.1 Local non-SNA 3270 Devices

### *5.1.1 Function*

The local non-SNA 3270 device statements are used to define terminals to the Hercules configuration. There are no required arguments for this particular device type but there are several optional arguments that may be specified.

To use this device a tn3270 client must connect to the host machine via the port number specified on the CNSLPORT statement. A valid tn3270 device type such as IBM-3278 must be used.

If the tn3270 client software allows it to specify a device type suffix (e.g. *IBM-3278@GROUPNAME*) then the suffix can be used to connect to that specific device number, if defined. If no suffix is used then the tn3270 client will be connected to the first available and defined 3270 device.

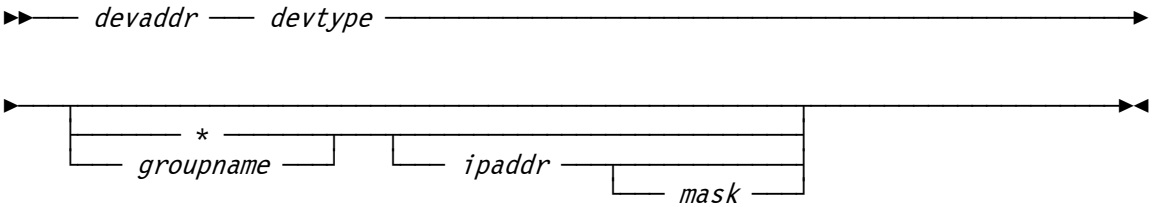
If a specific terminal device address is specified via the device type suffix of the tn3270 client software then it must be eligible to connect at that device address or the connection is immediately rejected. An alternative terminal device for which the tn3270 client might be eligible will not be automatically selected.

### *5.1.2 Syntax*

**Descriptive**

*devaddr devtype* [{*groupname* | ***} [*ipaddr* [*mask*]]]

**Diagram**




### *5.1.3 Parameter*

**devaddr** This is the device address.

**devtype** This is the device type. Valid terminal device types are 3270 and 3278.

**groupname** If a terminal group name is given on the device statement then a device type suffix with this group name can be used to indicate that a device in this group is to be used. If a group name is specified as a terminal type suffix (e.g. IBM-3278@GROUPNAME) and there are no devices defined with that group name or no available devices remaining in

that group then the connection is rejected. If no group name is specified as a terminal type suffix then the connection will only be eligible to terminal devices which also have no group name specified.

The terminal group name should be 1-8 alphanumeric characters in length, the first character being alphabetic and it should not be a hexadecimal number. Upper and lower case letters in the group name are considered to be equivalent.

***** The asterisk is used to indicate any terminal group name. It may be omitted if there are no additional arguments following the group name. If an IP address (and optionally mask) is specified and there is no specific group name desired then the asterisk must be coded to distinguish the terminal group name and the IP address arguments.

**ipaddr [mask]** The optional IP address and subnet masks specify the IP address(es) at which client(s) are allowed to connect via the device address identified by the device statement. This provides an additional means of specifying which device(s) a client tn3270 session may connect to.

If the IP address and mask (default 255.255.255.255 if not specified) of the tn3270 client attempting to connect match the IP address/mask entered on the device statement, then the client is eligible to connect at this device address. Otherwise the client is ineligible to connect at this device address and the next available device, if any, for which the client is eligible to connect, will be attempted.

If no permissible terminal devices remain (terminal devices for which the client is eligible to connect) or there are no more available terminal devices then the client connection is rejected.

The optional IP address and subnet mask may be specified in conjunction with the terminal group argument. In this case the terminal group argument must be specified ahead of the optional IP address and subnet mask arguments. To specify an IP address and a subnet mask without also specifying a terminal group use an asterisk ("*") as the group name substitute or placeholder.

If an IP address / mask are not specified then any client tn3270 session is allowed to connect to the device (provided they are also a member of the specified terminal group, if any).

The terminal group name must match if specified, regardless of any optional IP address / mask. To summarize, the device number suffix always takes precedence over any group name and any group name always takes precedence over any IP address/mask value.

### *5.1.4 Examples*

**Example 1:**

Define a 3270 device on device address 0200.

```
0200 3270
```

**Example 2:**

Define a 3270 device on device address 0200. Allow clients with any terminal group name but only from IP address 192.168.0.100 to connect.

```
0200 3270 * 192.168.0.100
```

**Example 3:**

Define a 3270 device on device address 0200. Allow clients with any terminal group name but only from IP address 192.168.0.100 with subnet mask 255.255.255.0 to connect.

```
0200 3270 * 192.168.0.100 255.255.255.0
```

**Example 4:**

Define a 3270 device on device address 0200. Allow only clients with terminal group name CONSOLE and from IP address 192.168.0.100 with subnet mask 255.255.255.0 to connect.

```
0200 3270 CONSOLE 192.168.0.100 255.255.255.0
```

**Example 5:**

Define a 3270 device on device address 0200. Allow only clients with terminal group name TSOTERM to connect.

```
0200 3270 TSOTERM
```

## 5.2 Integrated 3270 (SYSG) Console

### *5.2.1 Function*

The integrated 3270 (SYSG) console is similar to a local non-SNA 3270 device, except that it is not addressed by a subchannel number and it is supported only by certain system control programs. The SYSG console is defined like a 3270 device except that the device type is SYSG and the device address is ignored. Only one SYSG console can be defined in a configuration.

Use tn3270 client software to connect to the SYSG console device via the port number specified on the CNSLPORT statement, just as you would connect to a regular local non-SNA 3270 device.

The SYSG console configuration statement recognizes optional arguments which specify group name and IP address in the same way as previously described for a local non-SNA 3270 device. These optional arguments provide a means to ensure that a given tn3270 client can connect directly to the SYSG console.

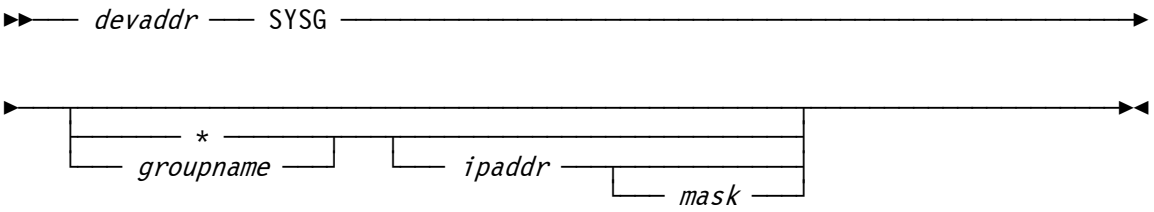
If the group name and IP address arguments are not specified, then the SYSG console is considered to be a member of the general pool of devices eligible for connection to any incoming tn3270 client.

### *5.2.2 Syntax*

**Descriptive**

*devaddr* SYSG [{*groupname* | *}] [*ipaddr* [*mask*]]]

**Diagram**




### *5.2.3 Parameter*

**devaddr** This is the device address. In the case of the integrated 3270 (SYSG) console this address is ignored.

**SYSG** This is the device type. The only valid device type is SYSG.

**groupname** If a terminal group name is given on the device statement then a device type suffix with this group name can be used to indicate that a device in this group is to be used. If a group name is specified as a terminal type suffix (e.g. IBM-3278@GROUPNAME) and there are no devices defined with that group name or no available devices remaining in that group then the connection is rejected. If no group name is specified as a terminal type suffix then the connection will only be eligible to terminal devices which also have no group name specified.

The terminal group name should be 1-8 alphanumeric characters in length, the first character being alphabetic and it should not be a hexadecimal number. Upper and lower case letters in the group name are considered to be equivalent.

***** The asterisk is used to indicate any terminal group name. It may be omitted if there are no additional arguments following the group name. If an IP address (and optionally mask) is specified and there is no specific group name desired then the asterisk must be coded to distinguish the terminal group name and the IP address arguments.

**ipaddr [mask]** The optional IP address and subnet masks specify the IP address(es) at which client(s) are allowed to connect via the device address identified by the device statement. This provides an additional means of specifying which device(s) a client tn3270 session may connect to.

If the IP address and mask (default 255.255.255.255 if not specified) of the tn3270 client attempting to connect match the IP address/mask entered on the device statement, then the client is eligible to connect at this device address. Otherwise the client is ineligible to connect at this device address and the next available device, if any, for which the client is eligible to connect, will be attempted.

If no permissible terminal devices remain (terminal devices for which the client is eligible to connect) or there are no more available terminal devices then the client connection is rejected.

The optional IP address and subnet mask may be specified in conjunction with the terminal group argument. In this case the terminal group argument must be specified ahead of the optional IP address and subnet mask arguments. To specify an IP address and a subnet mask without also specifying a terminal group use an asterisk ("*") as the group name substitute or placeholder.

If an IP address / mask are not specified then any client tn3270 session is allowed to connect to the device (provided they are also a member of the specified terminal group, if any).

The terminal group name must match if specified, regardless of any optional IP address / mask. To summarize, the device number suffix always takes precedence over any group name and any group name always takes precedence over any IP address/mask value.

### *5.2.4 Examples*

**Example 1:**

Define an integrated 3270 (SYSG) console on device address 0000 (the device address is ignored for SYSG consoles).

```
0000 SYSG
```

**Example 2:**

Define an integrated 3270 (SYSG) console on device address 0000. Allow only clients with terminal group name SYSGCONS and from IP address 192.168.0.100 with subnet mask 255.255.255.0 to connect.

```
0000 SYSG SYSGCONS 192.168.0.100 255.255.255.0
```

**Example 3:**

Define an integrated 3270 (SYSG) console on device address 0000. Allow clients with any terminal group name but only from IP address 192.168.0.100 to connect.

```
0000 SYSG * 192.168.0.100
```

**Example 4:**

Define an integrated 3270 (SYSG) console on device address 0000. Allow only clients with terminal group name SYSGCONS and from IP address 192.168.0.100 to connect.

```
0000 SYSG SYSGCONS 192.168.0.100
```

**Example 5:**

Define an integrated 3270 (SYSG) console on device address 0000. Allow only clients with terminal group name SYSGCONS to connect.

```
0000 SYSG SYSGCONS
```

## 5.3 Console Printer-Keyboard Devices

### *5.3.1 Function*

These device statements are used to define “Console Printer-Keyboard” devices to Hercules. To use the Console Printer-Keyboard device a telnet client must connect to the host machine via the port number specified on the CNSLPORT statement in that configuration file.

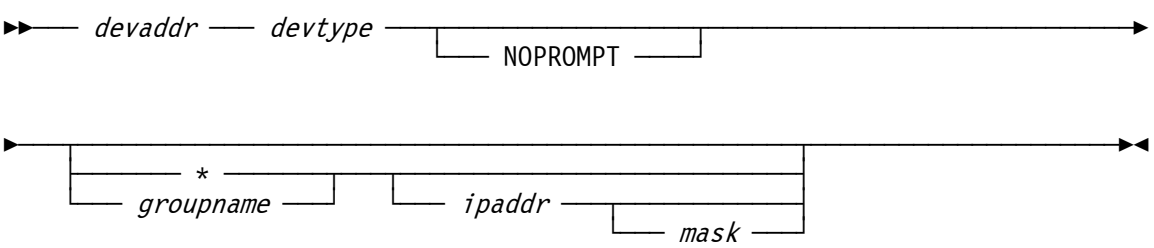
If the telnet client software allows it to specify a device type suffix (e.g. ansi@0009) then the suffix can be used to specify the specific 1052 or 3215 device to which the client should connect. If no suffix is used in the telnet client software (or the software does not allow it), then the client will be connected to the first available 1052 or 3215 device for which it is eligible.

### *5.3.2 Syntax*

**Descriptive**

*devaddr devtype* [NOPROMPT] [{*groupname* | *}] [*ipaddr* [*mask*]]]

**Diagram**




### *5.3.3 Parameter*

**devaddr** This is the device address.

**devtype** This is the device type. Valid terminal device types are 1052 and 3215.

**NOPROMPT** The NOPROMPT argument may be specified to cause suppression of the “ENTER INPUT FOR CONSOLE DEVICE nnnn” prompt message which is otherwise normally issued to the device whenever the system is awaiting input from the device.

The NOPROMPT argument, if specified, must preceed the optional group name and any ipaddr mask arguments.

**groupname** If a terminal group name is given on the device statement then a device type suffix with this group name can be used to indicate that a device in this group is to be used. If a group name is specified as a terminal type suffix (e.g. IBM-3278@GROUPNAME) and there are no devices defined with that group name or no available devices remaining in that group then the connection is rejected. If no group name is specified as a terminal type suffix then the connection will only be eligible to terminal devices which also have no group name specified.

The terminal group name should be 1-8 alphanumeric characters in length, the first character being alphabetic and it should not be a hexadecimal number. Upper and lower case letters in the group name are considered to be equivalent.

***** The asterisk is used to indicate any terminal group name. It may be omitted if there are no additional arguments following the group name. If an IP address (and optionally mask) is specified and there is no specific group name desired then the asterisk must be coded to distinguish the terminal group name and the IP address arguments.

**ipaddr [mask]** The optional IP address and subnet masks specify the IP address(es) at which client(s) are allowed to connect via the device address identified by the device statement. This provides an additional means of specifying which device(s) a client tn3270 session may connect to.

If the IP address and mask (default 255.255.255.255 if not specified) of the tn3270 client attempting to connect match the IP address/mask entered on the device statement, then the client is eligible to connect at this device address. Otherwise the client is ineligible to connect at this device address and the next available device, if any, for which the client is eligible to connect, will be attempted.

If no permissible terminal devices remain (terminal devices for which the client is eligible to connect) or there are no more available terminal devices then the client connection is rejected.

The optional IP address and subnet mask may be specified in conjunction with the terminal group argument. In this case the terminal group argument must be specified ahead of the optional IP address and subnet mask arguments. To specify an IP address and a subnet mask without also specifying a terminal group use an asterisk ("*") as the group name substitute or placeholder.

If an IP address / mask are not specified then any client tn3270 session is allowed to connect to the device (provided they are also a member of the specified terminal group, if any).

The terminal group name must match if specified, regardless of any optional IP address / mask. To summarize, the device number suffix always takes precedence over any group name and any group name always takes precedence over any IP address/mask value.

### *5.3.4 Examples*

**Example 1:**

Define a 1052 console printer-keyboard device on device address 0009.

```
0009 1052
```

**Example 2:**

Define a 1052 console printer-keyboard device on device address 0009 and suppress the "ENTER INPUT FOR CONSOLE DEVICE nnnn" prompt messages.

```
0009 1052 NOPROMPT
```

**Example 3:**

Define a 3215 console printer-keyboard device on device address 0009. Allow only clients with terminal group name CONSOLES and from IP address 192.168.0.100 to connect.

```
0009 3215 CONSOLES 192.168.0.100 255.255.255.0
```

**Example 4:**

Define a 3215 console printer-keyboard device on device address 0009. Allow only clients with terminal group name CONSOLES and from IP address 192.168.0.100 with subnet mask 255.255.255.0 to connect. Additionally suppress the “ENTER INPUT FOR CONSOLE DEVICE nnnn” prompt messages.

```
0009 3215 NOPROMPT CONSOLES 192.168.0.100 255.255.255.0
```

**Example 5:**

Define a 3215 console printer-keyboard device on device address 0009. Allow clients with any terminal group name but only from IP address 192.168.0.100 with subnet mask 255.255.255.0 to connect. Additionally suppress the “ENTER INPUT FOR CONSOLE DEVICE nnnn” prompt messages.

```
0009 3215 NOPROMPT * 192.168.0.100 255.255.255.0
```

## 5.4 Integrated Console Printer-Keyboard Devices

### *5.4.1 Function*

These device statements are used to define “Integrated Console Printer-Keyboard” devices to the Hercules configuration. The statements have one optional argument, the default command prefix for sending input to the device. The default command prefix is “/”. All integrated devices must use a different command prefix.

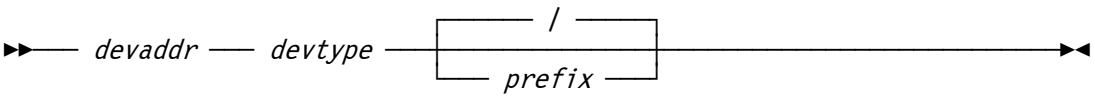
To send a logon command to a 1052-C or 3215-C enter “/logon” on the Hercules console.

### *5.4.2 Syntax*

**Descriptive**

*devaddr devtype* [{*prefix* | /}]

**Diagram**




### *5.4.3 Parameter*

**devaddr** This is the device address.

**devtype** This is the device type. Valid terminal device types are 1052-C and 3215-C.

**prefix** This is the command prefix for the device. The default command prefix is “/”. All devices must use a different command prefix.

### *5.4.4 Examples*

**Example 1:**

Define a 3215-C integrated console printer-keyboard device on device address 0009. Specify "/" as command prefix.

```
0009 3215-C /
```

## 5.5 Card Reader Devices

### *5.5.1 Function*

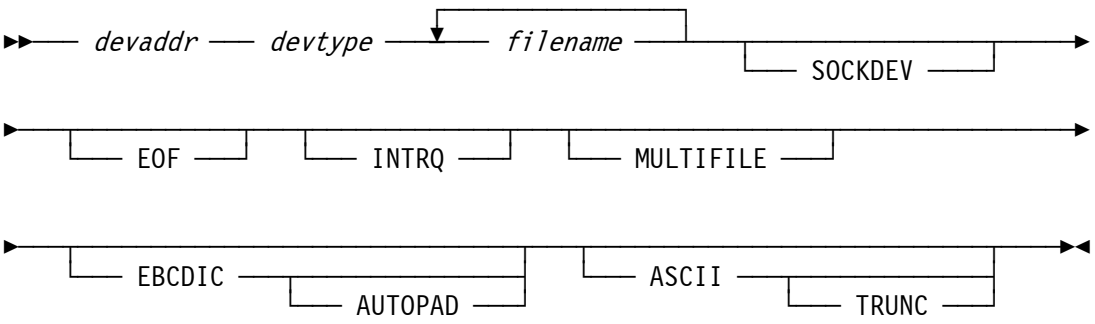
The card reader device statements are used to define card readers to the Hercules configuration. The argument specifies a list of file names containing card images. Additional arguments can be defined after the file names.

### *5.5.2 Syntax*

**Descriptive**

*devaddr devtype filename* [*filename* **...** ] [SOCKDEV] [EOF] [INTRQ]
[MULTIFILE] [EBCDIC [AUTOPAD]] [ASCII [TRUNC]]

**Diagram**




### *5.5.3 Parameter*

**devaddr** This is the device address.

**devtype** This is the device type. Valid card reader types are 1442, 2501, 3505.

**filename** The name (and optionally the path) of the card reader file.

**SOCKDEV** Indicates that the card reader is a socket device, wherein the filename is actually a socket specification instead of a device filename. When used, there must be only one filename specified in the form "*port*", "*host:port*" or "*sockpath/sockname*". The device then accepts remote connections on the given TCP/IP port or Unix domain socket and reads data from the socket instead from a device file. This allows automatic remote submission of card reader data. See chapter "Submitting Jobs via the Socket Reader" for more details.

**EOF** EOF specifies that unit exception status is presented after reading the last card in the file. This option is persistent and will remain in effect until the reader is reinitialized with the INTRQ option.

**INTRQ** INTRQ specifies that unit check status with intervention required sense bytes is presented after reading the last card in the file. This option is persistent and will remain in effect until the reader is reinitialized with the EOF option.

**MULTIFILE** Specifies, when mutliple files are entered, to automatically open the next input file and continue reading whenever EOF is encountered on a given file. If not specified reading stops once EOF is reached on a given file and an attention interrupt is required to open and begin reading the next file.

**EBCDIC** Specifies that the file contains fixed length 80-byte EBCDIC records with no line-end delimiters.

**ASCII** Specifies that the file contains variable length lines of ASCII characters delimited by LF (line feed) or CRLF (carriage return line feed) sequences at the end of each line.

If neither EBCDIC or ASCII is coded then the device handler attempts to detect the format of the card image file when the device is first accessed. Auto-detection is not supported for socket devices and the default is ASCII if SOCKDEV is specified.

**TRUNC** With ASCII files, TRUNC defines, that lines longer than 80 characters are truncated instead of producing a unit check error.

**AUTOPAD** With EBCDIC files, AUTOPAD defines, that the file is automatically padded to a multiple of 80 bytes if necessary.

## *5.5.4 Examples*

**Example 1:**

Define a 3505 card reader device on device address 000C. The name and path of the card reader file is "D:/JCL/DUMMY.JCL". A unit exception status has to be presented after reading the last card in the file.

```
000C 3505 D:/JCL/DUMMY.JCL EOF
```

**Example 2:**

Define a 1442 card reader device on device address 000C. The name and path of the card reader file is "D:/JCL/JCL.TXT". The file contains fixed length 80-byte EBCDIC records, automatically padded to a multiple of 80 bytes if necessary. A unit check status with intervention required sense bytes has to be presented after reading the last card in the file.

```
000C 1442 D:/JCL/JCL.TXT EBCDIC AUTOPAD INTRQ
```

**Example 3:**

Define a 2501 card reader device on device address 000C. The card reader is a socket device accepting remote connections from address 127.0.0.1 port 2501. The file contains variable length lines of ASCII characters delimited by line feed or carriage return line feed sequences at the end of each line. Lines longer than 80 bytes have to be truncated and a unit exception status has to be presented after reading the last card in the file.

```
000C 2501 127.0.0.1:2501 SOCKDEV ASCII TRUNC EOF
```

## 5.6 Card Punch Devices

### *5.6.1 Function*

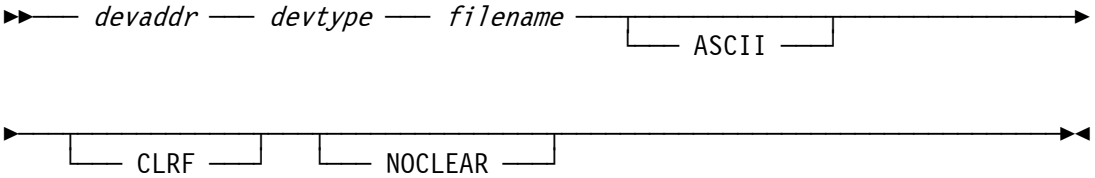
The card punch statement defines a card punch device to the Hercules configuration. The argument specifies the name of the file to which the punched output will be written. Additional arguments may be specified after the filename.

### *5.6.2 Syntax*

**Descriptive**

*devaddr devtype filename* [ASCII] [CRLF] [NOCLEAR]

**Diagram**




### *5.6.3 Parameter*

**devaddr** This is the device address.

**devtype** This is the device type. A valid card punch type is 3525.

**filename** The name (and optionally the path) of the card punch file.

**ASCII** ASCII specifies that the file will be written as variable length lines of ASCII characters, delimited by line feeds or carriage return line feed sequences. Trailing blanks are removed from each line. If the ASCII argument is not specified, the file is written as fixed length 80-byte EBCDIC records with no line-end delimiters.

**CRLF** This optional parameter specifies that carriage return line feed sequences are written at the end of each line. If the CRLF argument is not specified,line feeds are only written at the end of each line.

**NOCLEAR** This argument specifies that the output file will not be cleared to zero bytes when it is opened. If NOCLEAR is not specified, then any previous content of the file is destroyed when the file is opened for output.

### *5.6.4 Examples*

**Example 1:**

Define a 3525 card punch device on device address 000D. The punched output has to be written to the file "D/PCH/PCH1.TXT" as variable length lines of ASCII records delimited by carriage return line feed sequences. Any existing output in the file has to be kept when the file is opened for output.

```
000D 3525 D:/PCH/PCH1.TXT ASCII CRLF NOCLEAR
```

**Example 2:**

Define a 3525 card punch device on device address 000D. The punched output has to be written to the file "D/PCH/PCH2.TXT" as fixed length 80-byte EBCDIC records with no line-end delimiters. Any existing output in the file will be destroyed when the file is opened for output.

```
000D 3525 D:/PCH/PCH2.TXT
```

## 5.7 Line Printer Devices

### *5.7.1 Function*

The line printer device statement defines a printer to the Hercules configuration. The argument specifies the name of a file to which the printer output will be written. The output is written in the form of variable length lines of ASCII characters, delimited by line feeds or carriage return line feed sequences. Trailing blanks are removed from each line. Carriage control characters are translated to blank lines or ASCII form feed characters. If the file exists it will be overwritten.

If the filename begins with the vertical bar '|' pipe character, then it is removed and the remainder of the filename is interpreted as a command line (the name of a program or batch file followed by any necessary arguments) to which to "pipe" the printer output to. This is known as the "print-to-pipe" feature. All printer output is then sent to the piped program's stdin input, and all of the piped program's stdout and stderr output is piped back to Hercules for displaying on the hardware console.

SOCKDEV indicates that the line printer is a socket device wherein the filename is actually a socket specification in the form *host:port* instead of a device filename. The device then accepts remote connections on the given TCP/IP port and writes data to the socket instead to a device file. A Windows software that supports a socket printer is Fish's "HercPrt" (Hercules Remote Printer Spooler). It spools the print output from a Hercules socket printer and creates either PDF, RTF or just plain text files. A control file enables HercPrt to split the printouts on job separator page boundaries and name the file according the the actual job accounting field values.

### *5.7.2 Syntax*

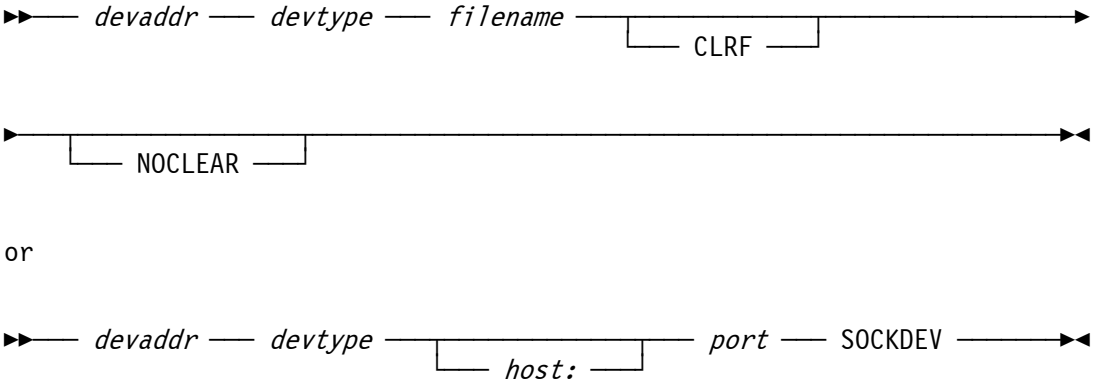
**Descriptive**

*devaddr devtype filename* [CRLF] [NOCLEAR]

or

*devaddr devtype* [*host:*]*port* SOCKDEV

**Diagram**



### 5.7.3 Parameter

**devaddr** This is the device address.

**devtype** This is the device type. Valid printer types are 1403 and 3211.

**filename** The name (and optionally the path) of the printer file or the “print-to-pipe” command line. If the filename begins with the vertical bar '|' pipe character, then it is removed and the remainder of the filename is interpreted as a command line (the name of a program or batch file followed by any necessary arguments) to which to "pipe" the printer output to. This is known as the "print-to-pipe" feature. All printer output is then sent to the piped program's stdin input, and all of the piped program's stdout and stderr output is piped back to Hercules for displaying on the hardware console.

If the "print-to-pipe" command line contains arguments, then quotes must be placed around the entire filename string including the vertical bar. If the "print-to-pipe" command line itself contains quotes, then the command line must be enclosed in apostrophes instead of quotes.

**CRLF** This optional parameter specifies that carriage return line feed sequences are written at the end of each line. If the CRLF argument is not specified then line feeds only are written at the end of each line.

**NOCLEAR** This argument specifies that the output file will not be cleared to zero bytes when it is opened. If NOCLEAR is not specified, then any previous content of the file is destroyed when the file is opened for output.

**SOCKDEV** SOCKDEV indicates that the line printer is a socket device wherein the filename is actually a socket specification instead of a device filename. When used, there must only be one filename specified in the form *port* or *host:port*. The device then accepts remote connections on the given TCP/IP port and writes data to the socket instead to a device file. This allows automatic remote spooling of line printer data. The sockdev option is mutually exclusive with all other printer options (like CRLF etc.) and must be specified alone.

### 5.7.4 Examples

**Example 1:**

Define a 1403 line printer device on device address 000E. The printed output has to be written to the file "D:\PRT\PRT1.TXT" with carriage return line feed sequences at the end of each line. Any existing output in the file has to be kept when the file is opened for output.

```
000E 1403 D:/PRT/PRT1.TXT CRLF NOCLEAR
```

**Example 2:**

Define a 3211 line printer device on device address 000F. The printed output has to be written to the file "D:\PRT\PRT2.TXT" with line feeds only at the end of each line. Any existing output in the file will be overwritten when the file is opened for output.

```
000F 3211 D:/PRT/PRT2.TXT
```

**Example 3:**

Define a 1403 line printer device on device address 000E. The line printer is a socket device with the IP address 192.168.0.199 and port 1403. Output is written to the socket instead to a device file.

```
000E 1403 192.168.0.199:1403 SOCKDEV
```

**Example 4:**

Define a 1403 line printer device on device address 000E. The command line for the print-to-pipe feature is “/usr/bin/lpr”, the argument for the ‘lpr’ program is “-Phplj” (Unix example). In the case of the Windows example the command line for the print-to-pipe feature is “C:\utils\pr”, the argument for the ‘pr’ program is “-PLPTP1:” . Each printed line will have a carriage return line feed sequence at the end.

```
000E 1403 "|/usr/bin/lpr -Phplj" crlf         (for Unix)

000E 1403 "|c:\utils\pr -s -PLPT1:" crlf      (for Windows)
```

## 5.8 Emulated Tape Devices

Tape device statements are used to define tape devices to the Hercules configuration. Five types of emulation are supported:

- SCSI tapes
- Optical Media Attach (OMA) virtual files
- AWSTAPE virtual files
- HET virtual files
- Fake Tape virtual files

### *5.8.1 SCSI Tapes*

#### 5.8.1.1 Function

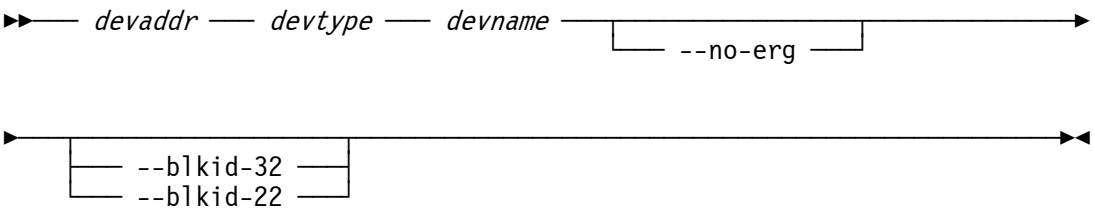
When defining SCSI tapes the argument specifies the tape device name (usually **/dev/nst0**). SCSI tapes are read and written using variable length EBCDIC blocks and filemarks exactly like a mainframe tape volume (see also the AUTO_SCSI_MOUNT system parameter).

#### 5.8.1.2 Syntax

**Descriptive**

*devaddr devtype devname* [--no-erg] [--blkid-32 | --blkid-22]

**Diagram**




#### 5.8.1.3 Parameter

**devaddr** This is the device address.

**devtype** This is the device type. Valid device types are 3410, 3420, 3422, 3430, 3480, 3490, 8809 and 9347.

**devname** The tape device name (usually "**/dev/nst0**" on Linux or "**\\.\Tape0**" on Windows).

**--no-erg** This option is intended to prevent issuance of the Erase Gap command to those SCSI tape drives that do not support it (e.g. Quantum DLT series). It causes Hercules's device emulation logic to ignore any Erase Gap commands issued to the drive and to return immediate 'success' instead.

This option should only be used (specified) for drives such as the Quantum which support switching from read mode to write mode in the middle of a data stream without the need of an intervening Erase Gap command. Specifying it for any other model SCSI drive may cause incorrect functioning as a result of the Erase Gap command not being issued to the actual SCSI hardware.

Check the manufacturer information for your particular model of SCSI-attached tape drive (and/or use Fish's "ftape" Windows utility) to determine whether or not this option is needed for your particular drive.

**--blkid-32** This option indicates that your SCSI-attached tape drive only supports 32-bit block-ids (as used by 3590 drives) and not the 22-bit format used by 3480/3490 drives. You should only specify this option if you intend to define the drive as a model 3480 or 3490 device and then only if your actual SCSI drive uses 32-bit block-ids. If you define your Hercules tape drive as a model 3590 device however this option is not required as 3590 drives are already presumed to use 32-bit block-ids.

Specifying this option on a 3480/3490 device statement will cause Hercules device emulation logic to automatically translate the actual SCSI tape drive's 32-bit block-id into 22-bit format before returning it back to the guest operating system (since that is the format it expects it to be in for a model 3480/3490 drive), and to translate the guest's 22-bit format block-id into 32-bit format before sending it to the actual SCSI hardware (since that is the format that the actual hardware requires it to be in).

**--blkid-22** The opposite of the above --blkid-32 option.

#### 5.8.1.4 Examples

**Example 1:**

Define a 3420 tape device on device address 0580. The tape device is a SCSI-attached tape drive which only supports 32-bit block-ids.

```
0580 3420 /dev/nst0 --blkid-32
```

### *5.8.2 Optical Media Attach (OMA) virtual files*

#### 5.8.2.1 Function

OMA device statements are used to define Optical Media Attach (OMA) virtual files to the Hercules configuration. OMA virtual files are read-only files which normally reside on CDROM. OMA virtual tapes consist of one CDROM file corresponding to each physical file of the emulated tape. An ASCII text file called the “Tape Descriptor File” (TDF) specifies the names of the files which make up the virtual tape. The argument specifies the name of the tape descriptor file.

Each file on the virtual tape can be in one of the following three formats:

**TEXT** TEXT files consist of variable length ASCII records delimited by carriage return line feed (CRLF) sequences at the end of each record. Each record is translated to EBCDIC and presented to the program as one physical tape block.

**FIXED nnnnn** FIXED files consist of fixed length EBCDIC blocks of the specified length (nnnnn).

**HEADERS** HEADERS files consist of variable length EBCDIC blocks. Each block is preceded by a 12-byte header.

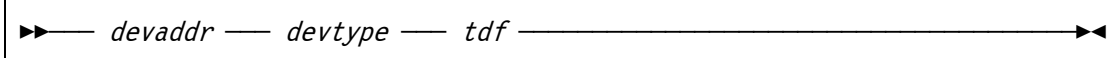
If you have any IBM manuals in BookManager format on CDROM, you can see some examples of TDF files in the “\TAPES” directory on the CDROM.

#### 5.8.2.2 Syntax

**Descriptive**

*devaddr devtype tdf*

**Diagram**




#### 5.8.2.3 Parameter

**devaddr** This is the device address

**devtype** This is the device type Valid device types are 3410, 3420, 3422, 3430, 3480, 3490, 8809 and 9347.

**tdf** The filename (and path) of the tape descriptor file.

#### 5.8.2.4 Examples

**Example 1:**

Define a 3480 tape device on device address 0582. The tape device is an Optical Media Attach (OMA) virtual file (CD-ROM). The filename of the Tape Descriptor File (TDF) is "UAA186.TDF" located in path "/CDROM/TAPES/".

```
0582 3480 /CDROM/TAPES/UAA186.TDF
```

### *5.8.3 AWSTAPE virtual files*

#### 5.8.3.1 Function

AWSTAPE device statements are used to define AWSTAPE virtual files to the Hercules configuration. AWSTAPEs contain a complete tape in one file. AWSTAPE files consist of variable length EBCDIC blocks. Each block is preceded by a 6-byte header. Filemarks are represented by a 6-byte header with no data. This is the same format as is used by the IBM P/390 systems. The argument specifies the location of the AWSTAPE file.

#### 5.8.3.2 Syntax

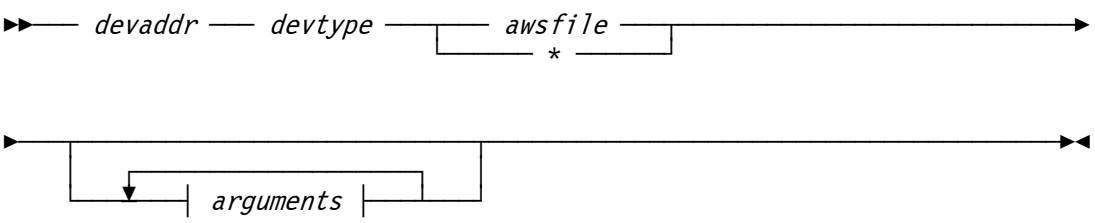
**Descriptive**

*devaddr devtype* {*awsfile* | ***} [*arguments*]

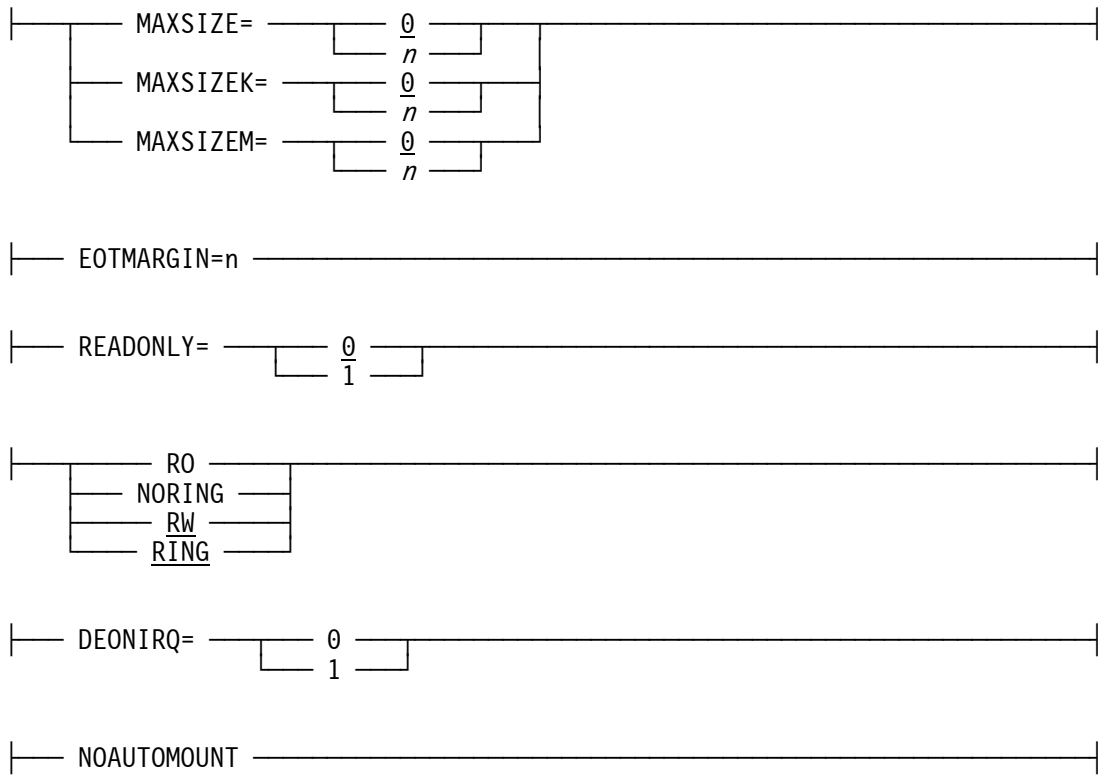
where arguments are:

[MAXSIZE={*n* | 0} | MAXSIZEK={*n* | 0} | MAXSIZEM={*n* | 0}]
[EOTMARGIN=*n*]
[READONLY={0 | 1}]
[RO | NORING | RW | RING]
[DEONIRQ={0 | 1}]
[NOAUTOMOUNT]

**Diagram**




where arguments are:



#### 5.8.3.3 Parameter

**devaddr** This is the device address.

**devtype** This is the device type. Valid device types are 3410, 3420, 3422, 3430, 3480, 3490, 8809 and 9347.

**awsfile** The filename (and path) of the AWSTAPE file or "*". An asterisk defines an empty tape station. The tape must be manually be loaded on request.

If the filename starts with the "@" character (at sign), the file then describes a list of tape emulation files to be loaded in succession. The syntax of each line in this file is identical to the information that can be coded directly after the device type in the overall configuration file. Any emulation file name parameter specified in this file may be substituded by the character "*", in which case it specifies a set of options to be applied to all additional emulation files specified in the list.

This function emulates an Automatic Cartridge Feeder (ACF. The ACF is a feature on cartridge type tape drives (3480, 3490, etc..) that automatically loads a new tape when a tape is removed from the drive. There is no real control over this device by the host as it just keeps on feeding tapes one after the other. Although the ACF feature is unique to cartridge type systems, the emulation accepts the use of the same technique for emulated 1/2 inch tapes reel drives as well.

Parameters are appended in succession. In all cases, if the same parameter is coded more than once, the last instance takes precedence. Therefore it is possible to specify a set of parameters in the base configuration file, another on an "*" line and another for each individual line in the list of files. Parameters are then appended in that order. A SCSI tape device should not be given in a file list.

The remaining parameters are described in section 5.8.6 ("Common parameters for AWS, HET and FakeTape virtual files").

#### 5.8.3.4 Examples

**Example 1:**

Define a 3490 tape device on device address 0580. The tape device is an AWSFILE virtual file named "R0673A.AWS" located in path "/S390/TAPES/".

```
0580 3490 /S390/TAPES/R0673A.AWS
```

### *5.8.4 HET virtual files*

#### 5.8.4.1 HET virtual files

HET device statements are used to define HET virtual files to the Hercules configuration. These contain a complete tape in one file and have the same structure as the AWSTAPE format with the added ability to have compressed data.

The first argument specifies the location of the HET file. The filename must end with ".HET" to be recognized by Hercules as a HET file (e.g. "023178.HET"). There are several additional arguments that control various HET settings.

### 5.8.4.2 Syntax

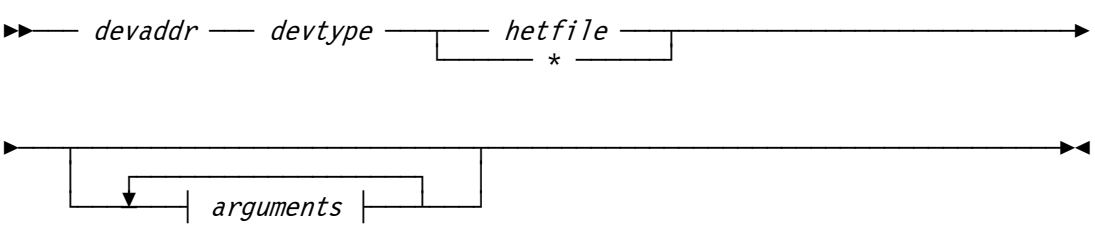
**Descriptive**

*devaddr devtype* {*hetfile* | ***} [*arguments*]

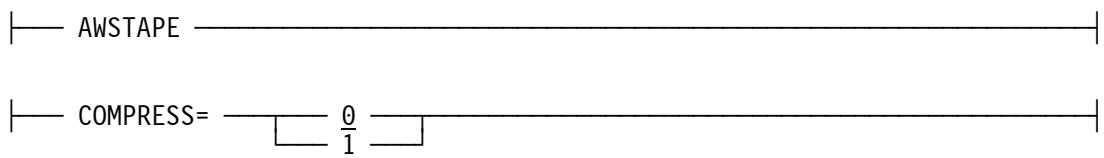
where arguments are:

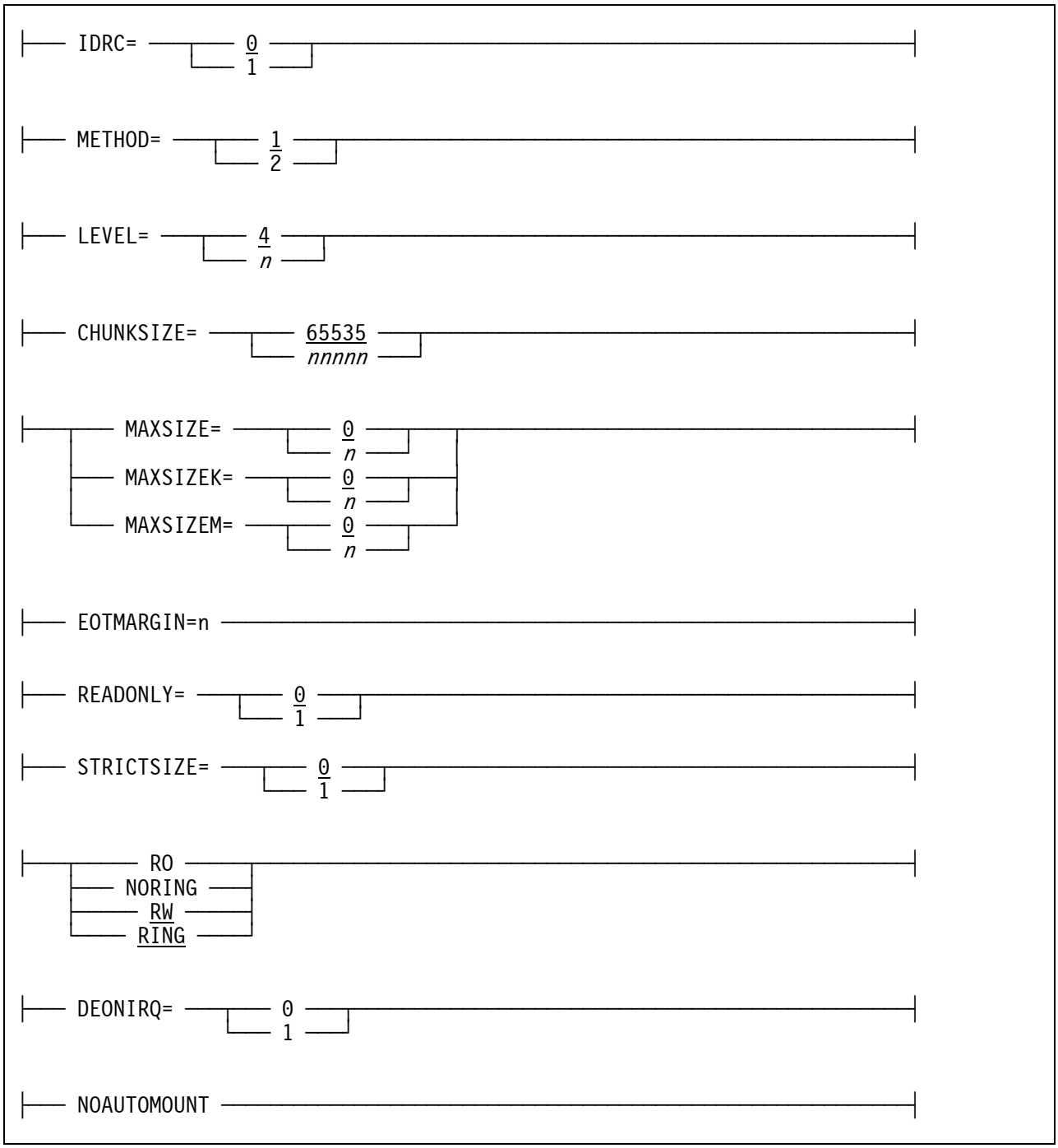
[AWSTAPE]
[COMPRESS={0 | 1}]
[IDRC={0 | 1}]
[METHOD={1 | 2}]
[LEVEL={*n* | 4}]
[CHUNKSIZE={*nnnnn* | 65535}]
[MAXSIZE={*n* | 0} | MAXSIZEK={*n* | 0} | MAXSIZEM={*n* | 0}]
[EOTMARGIN=*n*]
[READONLY={0 | 1}]
[STRICTSIZE={0 | 1}]
[RO | NORING | RW | RING]
[DEONIRQ={0 | 1}]
[NOAUTOMOUNT]

**Diagram**




where arguments are:






### 5.8.4.3 Parameter

**devaddr** This is the device address.

**devtype** This is the device type. Valid device types are 3410, 3420, 3422, 3430, 3480, 3490, 8809 and 9347.

**hetfile** The filename (and path) of the HET file or "*". An asterisk defines an empty tape station. The tape must be manually loaded on request.

If the filename starts with the "@" character (at sign), the file then describes a list of tape emulation files to be loaded in succession. The syntax of each line in this file is identical

to the information that can be coded directly after the device type in the overall configuration file. Any emulation file name parameter specified in this file may be substitued by the character "*", in which case it specifies a set of options to be applied to all additional emulation files specified in the list.

This function emulates an Automatic Cartridge Feeder (ACF). The ACF is a feature on cartridge type tape drives (3480, 3490, etc..) that automatically loads a new tape when a tape is removed from the drive. There is no real control over this device by the host as it just keeps on feeding tapes one after the other. Although the ACF feature is unique to cartridge type systems, the emulation accepts the use of the same technique for emulated 1/2 inch tapes reel drives as well.

Parameters are appended in succession. In all cases, if the same parameter is coded more than once, the last instance takes precedence. Therefore it is possible to specify a set of parameters in the base configuration file, another on an "*" line and another for each individual line in the list of files. Parameters are then appended in that order. A SCSI tape device should **not** be given in a file list!

**AWSTAPE** The AWSTAPE argument causes HET files to be written in AWSTAPE format. This disables the additional feature provided by the HET format.

**IDRC=n** COMPRESS and IDRC control whether compression should be used when writing to HET files. The value of *n* can be "1" to turn compression on (the default), or can be "0" to turn compression off. IDRC is currently a synonym for COMPRESS but may be used in the future to control another emulated tape drive feature. Therefore COMPRESS is the preferred method to turn compression on or off.

**COMPRESS=n** Same as above.

**METHOD=n** The METHOD argument allows you to specify which compression method to use. A value of "1" forces the use of ZLIB compression. A value of "2" forces the the use of BZIP2compression. The default is "1".

**LEVEL=n** The LEVEL option controls the level of compression. It ranges from "1" (for fastest compression) to "9" (for best compression). The default compression level is "4".

**CHUNKSIZE=nnnnn** The CHUNKSIZE option allows you to create HET files with different chunk sizes. The AWSTAPE (and therefore HET) formats allow each tape block to be logically broken up into smaller chunks. For instance, if the S/3x0 application creates tapes with a block size of 27998, those blocks would be broken down into *nnnnn* sized chunks. Although possible it is not recommended to change the chunk size, as decreasing this may reduce compression performance. The range of *nnnnn* is from 4096 to 65535. 65535 being the default.

**STRICTSIZE=n** Upon reaching the tape file size limit, depending on strictsize, the tape file will or will not be truncated to enforce the maxsize limit. The limit is only enforced during a write type operation. If the file already exists and the program only reads the file, then the file will not be truncated, regardless of the strictsize setting.

This affects any write that starts below the limit, but that would extend beyond the limit. The STRICTSIZE parameter only affects compressed HET files. On AWS tapes, the limit is always enforced, but the file is not truncated (the write does not occur, because first AWS tapes are never truncated and second the effects of the write are known in advance).

Regardless of strictsize, any write operation (Write, Write TM) will return a Unit Check with Equipment Check to the program if the file size exceeds the predefined limit. If STRICTSIZE is "0", the write will actually have been performed on the tape file. When STRICTSIZE is set to "1", the file will be truncated on the preceeding tape block boundary.

Care must be taken that regardless of the STRICTSIZE setting, the tape may become unusable for the guest program should such an event occur (absence of a Tape Mark for example). This option has no effect if maxsize is 0. The default is "0" (do not truncate).

The remaining parameters are described in section 5.8.6 (“Common parameters for AWS, HET and FakeTape virtual files”).

#### 5.8.4.4 Examples

**Example 1:**

Define a 3490 tape device on device address 0580. The tape device is a HET virtual file with name "R0674A.HET" located in path "/S390/TAPES/". BZIP2 compression has to be turned on at a level of 9 (best compression).

```
0580 3490 /S390/TAPES/R0674A.HET COMPRESS=1 METHOD=2 LEVEL=9
```

## *5.8.5 Fake Tape virtual files*

### 5.8.5.1 Function

Fake Tape virtual files contain a complete tape in one file. FakeTape files consist of variable length EBCDIC blocks. Each block is preceded by a 12-byte ASCII-hex-character header. Filemarks are re-presented by a 12-byte character header with no data. The FakeTape format is used by the Flex-ES system from Fundamental Software Inc (FSI). The argument specifies the location of the FakeTape file (for example “ickdsf.fkt”). "FLEX-ES" and "FakeTape" are trademarks of Fundamental Software, Inc.

### 5.8.5.2 Syntax

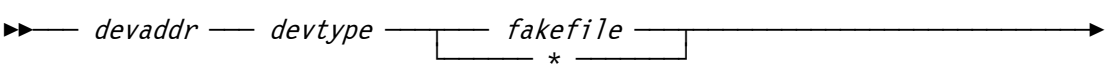
**Descriptive**

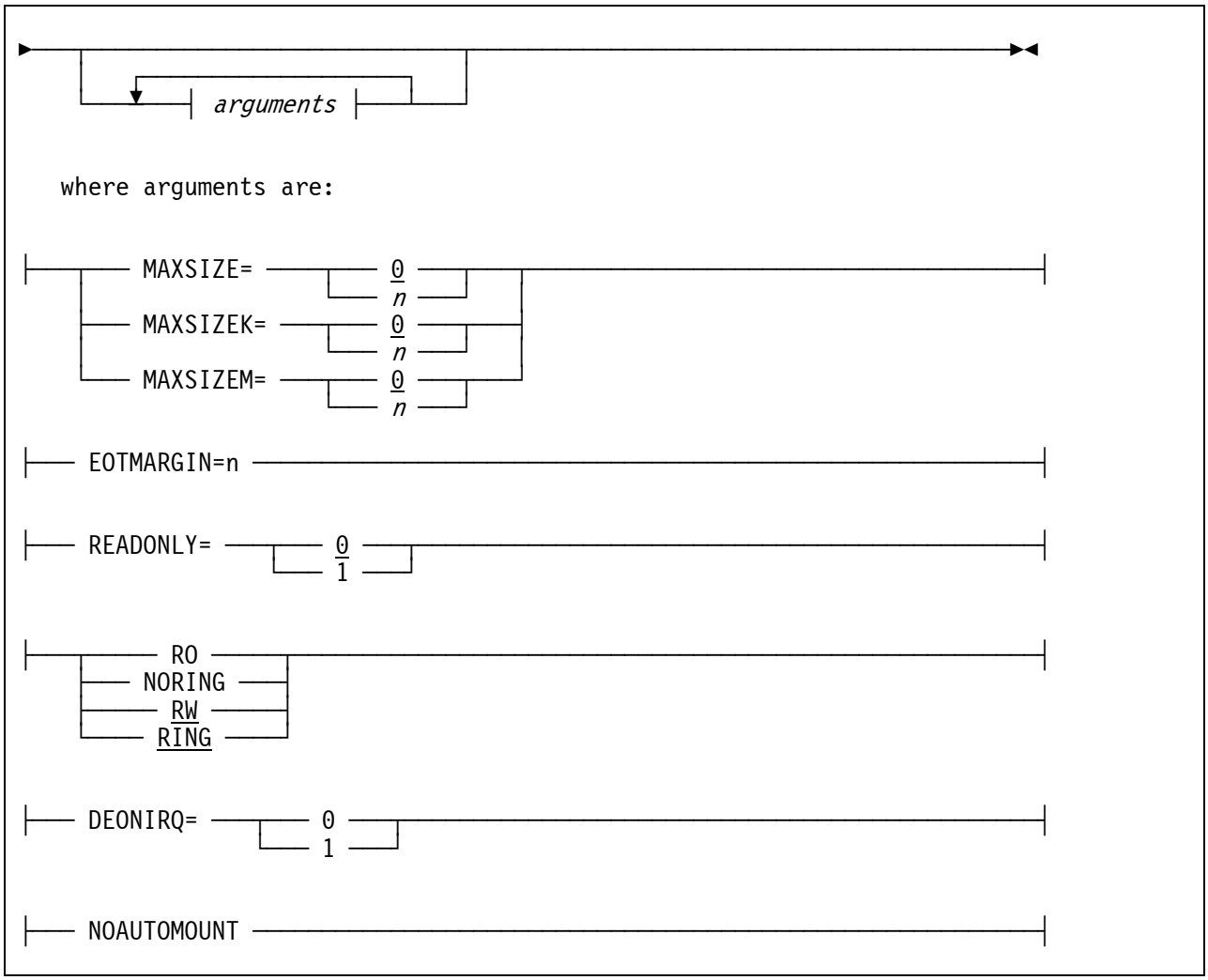
*devaddr devtype* {*fakefile* | ***} [*arguments*]

where arguments are:

[MAXSIZE={*n* | 0} | MAXSIZEK={*n* | 0} | MAXSIZEM={*n* | 0}]
[EOTMARGIN=*n*]
[READONLY={0 | 1}]
[RO | NORING | RW | RING]
[DEONIRQ={0 | 1}]
[NOAUTOMOUNT]

**Diagram**






### 5.8.5.3 Parameter

**devaddr** This is the device address.

**devtype** This is the device type. Valid device types are 3410, 3420, 3422, 3430, 3480, 3490, 8809 and 9347.

**fakefile** The filename (and path) of the Fake Tape virtual file or "*". An asterisk defines an empty tape station. The tape must be manually be loaded on request.

If the filename starts with the "@" character (at sign), the file then describes a list of tape emulation files to be loaded in succession. The syntax of each line in this file is identical to the information that can be coded directly after the device type in the overall configuration file. Any emulation file name parameter specified in this file may be substituded by the character "*", in which case it specifies a set of options to be applied to all additional emulation files specified in the list.

This function emulates an Automatic Cartridge Feeder (ACF. The ACF is a feature on Cartridge type tape drives (3480, 3490, etc..) that automatically loads a new tape when a tape is removed from the drive. There is no real control over this device by the host as it just keeps on feeding tapes one after the other. Although the ACF feature is unique to cartridge type systems, the emulation accepts the use of the same technique for emulated 1/2 inch tapes reel drives as well.

Parameters are appended in succession. In all cases, if the same parameter is coded more than once, the last instance takes precedence. Therefore it is possible to specify a set of parameters in the base configuration file, another on an “*” line and another for each individual line in the list of files. Parameters are then appended in that order. A SCSI tape device should not be given in a file list.

The remaining parameters are described in section 5.8.6 (“Common parameters for AWS, HET and FakeTape virtual files”).

#### 5.8.5.4 Examples

**Example 1:**

Define a 3490 tape device on device address 0580. The tape device is an Fake Tape virtual file named "R0528A.FKT" located in path "/S390/TAPES/".

```
0580 3490 /S390/TAPES/R0528A.FKT
```

### *5.8.6 Common parameters for AWS, HET and Fake Tape virtual files*

**MAXSIZE=n** This specifies the maximum number of bytes (specified in Bytes) for the emulated file. This parameter defaults to “0”, meaning there is no limit on the file size.

**MAXSIZEK=n** This is the same as above but specified in Kilobytes.

**MAXSIZEM=n** This is the same as above but specified in Megabytes.

**EOTMARGIN=n** Specifies the number of bytes remaining (before reaching *MAXSIZEx*) at which point the tape device will signal the presence of the “End-of-Tape” marker, thus allowing the program to switch to the next tape.

**READONLY=n** Specifies whether the tape is mounted read-only (without a write ring or with the cartridge protect switch set to "write protect"). A parameter of 1 means read-only; a parameter of 0 means read-write. If READONLY=1, RO or NORING is not specified; READONLY=0 is the default. Note that READONLY=0 does not override the host system file permission settings for the underlying AWS or HET file. If the AWS or HET file is marked read-only, the tape will be mounted read-only despite specification of READONLY=0.

**RO / NORING** Specifies that the tape is mounted read-only (without a write ring or with the cartridge protect switch set to "write protect"). RO and NORING are equivalent to READONLY=1.

**RW / RING** Specifies that the tape should be mounted read-write, if possible. RW and RING are equivalent to READONLY=0. This is the default if RO, NORING or READONLY=1 is not specified. Note that RW and RING do not override the host system file permission settings for the underlying AWS or HET file. If the AWS or HET file is marked read-only, the tape will be mounted read-only despite specification of RW or RING.

**DEONIRQ=n** Specifies whether a device end is presented if intervention is required during tape motion. A parameter of 1 selects this option; a parameter of 0 turns it off.

**NOAUTOMOUNT** Indicates that support for guest-initiated automatic tape volume mounting is to always be disabled for this tape device. Automatic guest tape-mount support is automatically globally enabled for all virtual (non-SCSI) tape devices by default whenever an allowable automount directory is defined via the AUTOMOUNT configuration file statement or the automount console command. The NOAUTOMOUNT option allows you to specifically disable such support for a given device.

The automount feature enables software running in guest operating systems to automatically mount, unmount and/or query for themselves the host "virtual tape volume"

filename mounted on a tape drive, via the use of special CCW opcodes (0x4B Set Diagnose and 0xE4 Sense ID) without any intervention on the part of the Hercules operator. An example of such a program for DOS/VSE called TMOUNT is provided in the util subdirectory of the distributed source code.

This is a sticky option. When specified, automount support for the device remains disabled until the option is specifically removed via a devinit command without the option specified. This means if NOAUTOMOUNT is enabled for a device while global automount functionality is currently disabled (because no AUTOMOUNT statement was specified at Hercules startup), then automount functionality remains disabled for the device even should global automount functionality be later manually enabled via an automount console command.

When the 0x4B Set Diagnose CCW is used to auto-mount a virtual tape volume onto a given tape drive, an absolute (fully-qualified) pathname should normally always be specified, but need not be if a path relative to the currently defined "default allowable" automount directory is used instead.

The default allowable automount directory is always the first "allowable" directory that was defined, or else the current directory if no allowable directories were specifically defined. There is always a default allowable directory whenever any allowable or unallowable automount directories are defined.

Fully-resolved, absolute-full-path filenames are defined as being those which, for Windows, have a ':' (colon) in the second position or, for other host operating systems (e.g. Linux), have a '/' (slash) in the first position. Paths which start with a '.' (period) are considered relative paths and will always be appended to the currently defined default allowable automount directory, before being resolved into fully-qualified paths by the host system (i.e. only fully-resolved absolute pathnames are used in the performance of the actual automatic tape volume mount).

For example, if more than one allowable automount directory is defined and the volume wishing to be mounted happens to reside in the second one, then a fully-qualified absolute pathname should of course be specified (or else one that is relative to the default directory which happens to resolve to the desired file).

All attempts to automount host files in any defined "disallowable" directory (or any sub-directory thereof [or otherwise *not* within any defined "allowable" directory or subdirectory]) will be rejected. An error message is always issued in such cases, just as one is whenever a successful mount or unmount is performed.

A sample guest automount program called TMOUNT for the DOS/VSE operating system is provided in the "util" subdirectory of the distributed source code.

## *5.8.7 Basic ACF (Automatic Cartridge Feeder) Support*

### 5.8.7.1 Function

The ACF (Automatic Cartridge Feeder) is a feature on cartridge type tape drives (3480, 3490, etc.) that automatically loads a new tape when a tape is removed (ejected) from the drive. There is no real control over this functionality by the host as the device just keeps on feeding tapes one after the other.

Although the ACF features is unique to catridge type tape systems the emulation accepts to use the same technique for emulated ½ inch tapes reel drives as well. ACF is supported through a file that contains a list of files (emulated tapes or catridges) that will be loaded one after the other.

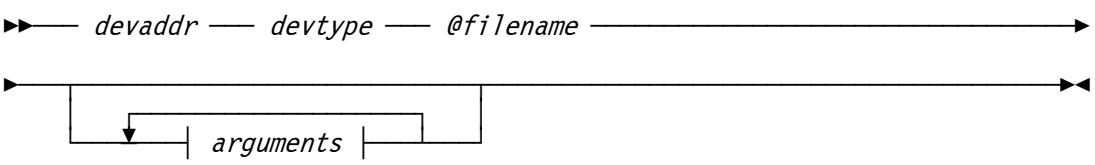
To manually reset the ACF to the top of the stack the DEVINIT panel command can be used to "reload" the ACF feature.

### 5.8.7.2 Syntax

**Descriptive**

*devaddr devtype @filename* [*arguments*]

**Diagram**




### 5.8.7.3 Parameter

**devaddr** This is the device address.

**devtype** This is the device type. Valid device types are 3410, 3420, 3422, 3430, 3480, 3490, 8809 and 9347.

**@filename** The filename (without the prefixing '@') contains a list of files that will be loaded one after the other.The filenames contained in the file list cannot describe another ACF file nor an SCSI tape handle (/dev/st*x*). However the files may be standard AWS, HET, Fake Tape or OMA files.

**arguments** These are the same arguments that apply to the AWS, HET, Fake Tape or OMA files, as described above.

### 5.8.7.4 Syntax ACF description file

```
# comment line
* arguments
filename [arguments]
```

### 5.8.7.5 Parameter

**#** The hash symbol starts a comment line.

***** Any options following the asterisk are applied to each file of the list, followed by the options specified on the device configuration entry, followed by the options specified on each individual entry of the file list. Care must be taken that all '*' lines are processed at once.

**filename** The filename (and path) of the AWS, HET, Fake Tape or OMA tape file.

**arguments** These are the same arguments that apply to the AWS, HET, Fake Tape or OMA files, as described above. The arguments are processed in the order in which they appear. Any conflicting argument overrides the previous one.

### 5.8.7.6 Examples

**Example 1:**

Define a 3490 Automatic Cartridge Feeder (ACF) on device address 0589. ZLIB compression is turned on. The file "newstack" describes a list of tape emulation files to be loaded in succession.

a) Hercules configuration file:

```
0589 3490 @NEWSTACK COMPRESS=1
```

b) “NEWSTACK” file:

```
* MAXSIZEM=16 EOTMARGIN=131072
tape01.aws COMPRESS=0
tape02.het MAXSIZEM=32 EOTMARGIN=65536
tape03.het MAXSIZE=0
```

The above example is equivalent to issuing the following (one at the start and one after each tape unload):

a) in the configuration file:

```
180 3420 tape01.aws maxsizeM=16 eotmargin=131072 compress=0
```

b) via devinit panel command:

```
devinit 180 tape02.het compress=1 maxsizeM=32 eotmargin=65536
devinit 180 tape03.het eotmargin=131072 compress=1 maxsize=0
```

**Example 2:**

Resolution of the arguments using ‘*’ lines.

The following ACF file

```
* compress=0
tape01.aws
* compress=1
tape02.aws
```

is equivalent to this one

```
* compress=1
tape01.aws
tape02.aws
```

# 5.9 Channel-to-Channel Adapters

## *5.9.1 Introduction*

The Channel-to-Channel Adapter device statements define CTC adapters to the Hercules configuration. All of the communications emulation implemented within Hercules use a CTCA (Channel-to-Channel Adapter) type device. Depending on the type, the CTCA device will provide either a point-to-point or a virtual network adapter interface to the driving system's TCP/IP stack or in the case of CTCT, a true CTCA connection to another instance of Hercules via a TCP/IP connection.

All current emulations, with the exception of VMNET and CTCT use the Universal TUN/TAP driver on Linux and TunTap32 (WinPCap) on the Windows platforms which creates a network interface on the driving system which allow Hercules to present frames to, and receive frames from the TCP/IP stack. This network interface is configured on Linux platforms by the hercifc program which is invoked by Hercules after the TUN/TAP device is opened.

The following are the emulation types currently supported:

- CTCI (Channel-to-Channel link to TCP/IP stack)
- CTCT (Channel-to-Channel emulation via TCP Connection)
- LCS (LAN Channel Station emulation)
- VMNET (Channel-to-Channel link via SLIP/VMNET)

## *5.9.2 CTCI (Channel-to-Channel link to TCP/IP stack)*

### 5.9.2.1 Function

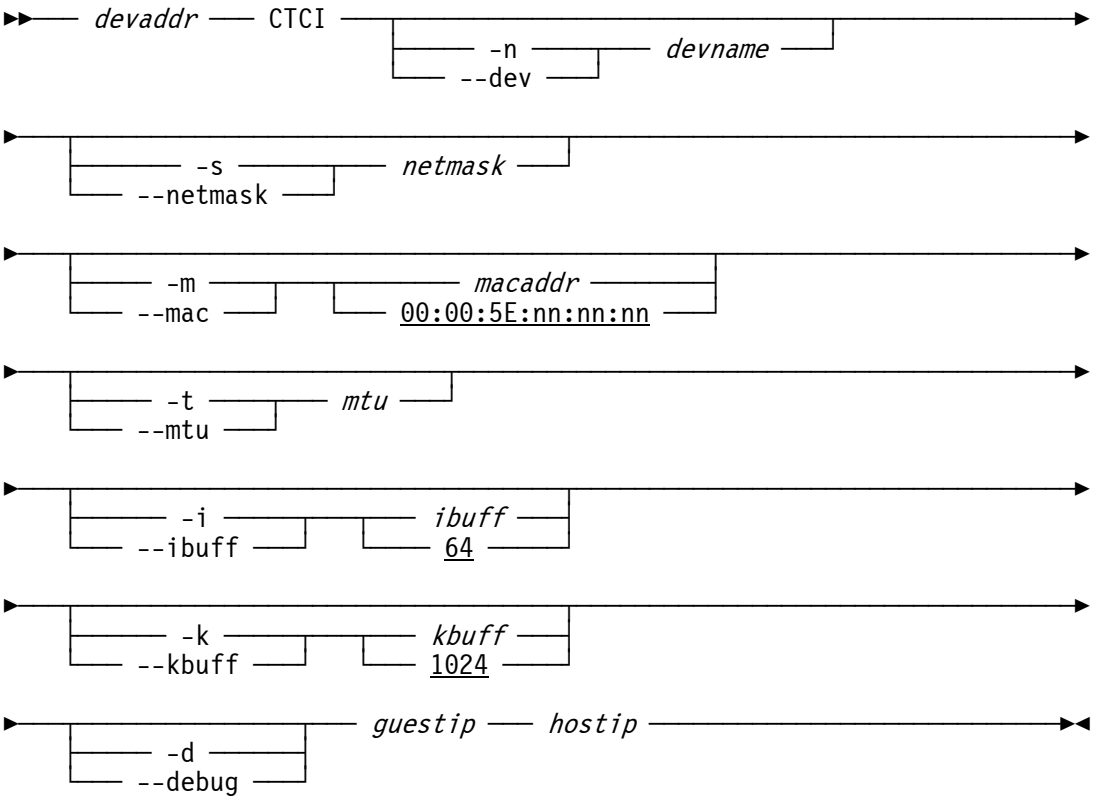
The CTCI is a point-to-point connection with the TCP/IP stack of the driving system on which Hercules is running.

### 5.9.2.2 Syntax

**Descriptive**

```
devaddr CTCI [{-n | --dev} devname]
             [{-s | --netmask} netmask]
             [{-m | --macaddr} macaddr | 00:00:5E:nn:nn:nn}]
             [{-t | --mtu} mtu | 1500]                          *Linux only
             [{-i | --ibuff} {ibuff | 64}]                      *Windows only
             [{-k | --kbuff} {kbuff | 1024}]                    *Windows only
             [-d | --debug]
             guestip hostip
```

**Diagram**




### 5.9.2.3 Parameter

**devaddr** This is the device address.

**CTCI** This specifies the device/protocol type.

**-n devname** For Linux: Specifies the name of the tunnel device to use. The default is "/dev/net/tun", which is correct for version 2.4 and above of the Linux kernel.

For Windows: Identifies the host network adapter. If your network adapter does not have a permanent (static) IP address assigned to it (e.g. you use DHCP and have a dynamic IP assigned) then instead of specifying an IP address as the second parameter in the device statement, you must specify the MAC address of the adapter. In this case the second argument (hostip) must be coded as "0.0.0.0". This is simply a placeholder because two IP addresses are expected to satisfy the device definition syntax.

**--dev devname** This is the same as "-n devname".

**-s netmask** This is the netmask to be configured on the link. Note: Since this is a point-to-point link netmask is meaningless from the perspective of the actual network device.

**--netmask netmask** This is the same as "-s netmask".

**--m macaddr** This is the optional hardware address of the interface in the format of either "xx:xx:xx:xx:xx:xx" or "xx-xx-xx-xx-xx-xx". The default is 00:00:5E:nn:nn:nn where the "*nn:nn:nn"* part is constructed from the last three octets of the specified guestip.

**--macaddr macaddr** This is the same as "-m macaddr".

**-t mtu** Linux only: Specifies the maximum transmission unit size (default 1500 bytes).

**--mtu mtu** This is the same as “-t mtu”.

**-i ibuff** Windows only: This specifies the TunTap32 I/O buffer size in KB. *nnnn* hast to be between 16 and 1024 (1 MB). The default value is 64 KB.

**--ibuff ibuff** This is the same as “-I ibuff”.

**-k kbuff** Windows only: This specifies the WinPcap device driver capture buffer size in KB. *nnnn* hast to be between 64 and 16384 (16 MB). The default value is 1024 (1 MB).

**--kbuff kbuff** This is the same as “-k kbuff”.

**-d** Specifies that debugging output has to be produced on the Hercules control panel. This should normally be left unspecified.

**--debug** This is the same as “-d”.

**guestip** Specifies the IP address of the guest operating system running under Hercules.

**hostip** Identifies the host network adapter to use. If the host system is configured with DHCP this should instead be the MAC address of the ethernet adapter you wish to have Hercules use to connect to the outside world. If this parameter is specified using the *-m* or *--dev* option, then the value here must be specified as "0.0.0.0" to satisfy the statement syntax.

#### 5.9.2.4 Examples

**Example 1:**

Define 3088 CTC adapters on device addresses 0E20 and 0E21. The device/protocol type is CTCI (Channel-to-Channel link to Linux TCP/IP stack). The IP address of the guest operating system running under Hercules is 192.168.1.99, the host IP address (the network adapter) to use is 192.168.1.100.

```
0E20 3088 CTCI 192.168.1.99 192.168.1.100
0E21 3088 CTCI 192.168.1.99 192.168.1.100
```

**Example 2:**

Define 3088 CTC adapters on device addresses 0E20 and 0E21. The device/protocol type is CTCI (Channel-to-Channel link to Win32 TCP/IP stack). The WinPcap device driver capture buffer has to be set to 2048 KB, the TunTap32 I/O buffer size has to be set to 128 KB. The IP address of the guest operating system running under Hercules is 192.168.1.99 and the MAC address of the network adapter to use is 00-80-B3-E1-DF-69.

```
0E20 3088 CTCI -n 00-80-B3-E1-DF-69 -k 2048 -i 128 192.168.1.99 0.0.0.0
0E21 3088 CTCI -n 00-80-B3-E1-DF-69 -k 2048 -i 128 192.168.1.99 0.0.0.0
```

### *5.9.3 CTCT (Channel-to-Channel emulation via TCP Connection)*

#### 5.9.3.1 Function

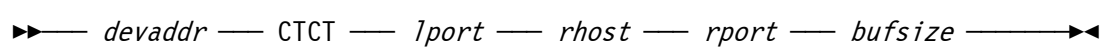
This is an emulated Channel-to-Channel adapter to another Hercules system. CTCT only supports IP traffic and cannot be used to transport NJE, SNA, PVM, etc. type payloads. This may change in the future.

### 5.9.3.2 Syntax

**Descriptive**

*devaddr* CTCT *lport rhost rport bufsize*

**Diagram**




### 5.9.3.3 Parameter

**devaddr** This is the device address.

**CTCT** This specifies the device/protocol type.

**lport** Specifies the local TCP port. This is the TCP port that Hercules will listen on for this CTC adapter.

**rhost** Specifies the remote host. This is the name or IP address of the remote host system that Hercules is running on, not the name or IP address of the OS running under this instance of Hercules.

**rport** Specifies the remote TCP port. The rport parameter on this system must match the lport parameter on the remote system and vice versa.

**bufsize** Specifies the buffer size for the link. If this link is used for IP traffic this parameter should be more than the MTU of the interface definition in the OS TCPIP stack.

### 5.9.3.4 Examples

**Example 1:**

Define 3088 CTC adapters on device addresses 0400 and 0401. The device/protocol type is CTCT (Channel-to-Channel emulation via TCP connection). The local TCP ports on which Hercules will listen are 30880 and 30881. The IP address of the remote host is 192.168.100.2, the remote ports are also 30880 and 30881. The buffer size for the link is 2048 KB.

```
0400 3088 CTCT 30880 192.168.100.2 30880 2048
0401 3088 CTCT 30881 192.168.100.2 30881 2048
```

## *5.9.4 CTCE (Enhanced Channel-to-Channel emulation via TCP Connection)*

### 5.9.4.1 Function

The CTCE device type will emulate a real 3088 Channel-to-Channel Adapter also for non-IP traffic, enhancing the CTCT capabilities. CTCE connections are also based on TCP/IP between two (or more) Hercules instances and require an even-odd pair of port numbers per device side.

Only the even port numbers are to be configured; the odd numbers are just derived by adding 1 to the configured even port numbers. The socket connection pairs cross-connect, the arrows are showing the the send -> receive direction:

```
InstanceA-lport-even -> InstanceB-rport-odd
InstanceB-lport-odd  <- InstanceA-rport-even
```

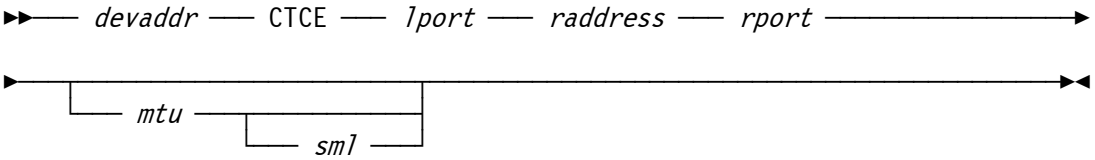
CTCE connected Hercules instances can be hosted on either Unix or Windows platforms, both sides do not need to be the same.

### 5.9.4.2 Syntax

**Descriptive**

*devaddr* CTCE *lport raddress rport* [*mtu* [*sml*]]

**Diagram**




### 5.9.4.3 Parameter

**devaddr** This is the device address.

**CTCE** This specifies the device/protocol type.

**lport** Specifies the even TCP/IP port on the local system.

**raddress** Specifies the TCP/IP address of the remote system.

**rport** Specifies the even TCP/IP port on the remote system.

**mtu** Optional MTU buffer size. The default size is 32778.

**sml** Optional small minimum for the MTU buffer size. The default size is 8.

### 5.9.4.4 Examples

**Example 1:**

Define a 3088 CTC connection with the CTCE protocol between two Hercules instances. Instance A is using address 192.168.1.100, Instance B is using address 192.168.1.200.

Definitions on Hercules Instance A with IP address 192.168.1.100:

```
0E40 CTCT 30880 192.168.1.200 30880
0E41 CTCT 30882 192.168.1.200 30882
```

Definitions on Hercules Instance B with IP address 192.168.1.200:

```
0E40 CTCT 30880 192.168.1.100 30880
0E41 CTCT 30882 192.168.1.100 30882
```

## *5.9.5 LCS (LAN Channel Station)*

### 5.9.5.1 Function

This statement defines an emulated LAN Channel Station adapter. This emulation mode appears to the operating system running in the Hercules machine as an IBM 8232 LCS device, an IBM 2216 router, an IBM 3172 running ICP (Interconnect Communications Program), the LCS3172 driver of an IBM P/390, or an IBM Open Systems Adapter (OSA).

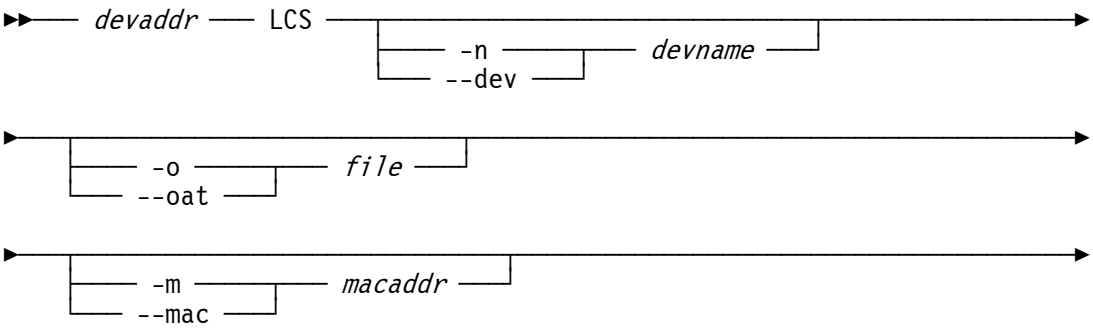
Rather than a point-to-point link, this emulation creates a virtual ethernet adapter through which the guest operating system running in the Hercules machine can communicate. As such, this mode is not limited to TCP/IP traffic, but in fact will handle any ethernet frame.

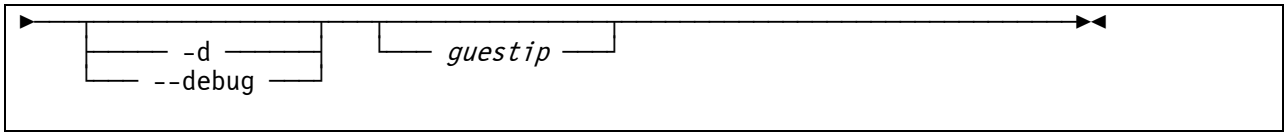
### 5.9.5.2 Syntax

**Descriptive**

```
devaddr LCS [{-n | --dev} devname]
            [{-o | --oat} file]
            [{-m | --mac} macaddr]
            [-d | --debug]
            [guestip]
```

**Diagram**






### 5.9.5.3 Parameter

**devaddr** This is the address pair of the LCS device. This pair must be an even-odd address. This can be specified as two separate LCS statements with a single device address ("0E20 LCS …") or as a single statement with a device address count ("0E20.2 LCS …").

**LCS** This specifies the LAN Channel Station device/protocol type.

**-n devname** Identifies the host network adapter to use:
- On Linux systems this is name of the TUN/TAP special character device, normally /dev/net/tun.
- On Windows this is either the IP or MAC address of the host systems network card. TunTap32 will automatically select the first network adapter it finds if the option is omitted, this may be not desirable for some users.

**--dev devname** This is the same as "-n devname".

**-o file** *file* specifies the filename of the Address Translation file. If this option is specified the optional *--mac* and *guestip* entries are ignored in preference to statements in the OAT file. For a description of the OAT file see next section.

**--oat file** This is the same as "-o file".

**-m macaddr** *macaddr* is the optional hardware address of the interface in the format xx:xx:xx:xx:xx:xx. If you use the -o/--oat option do not specify an address here, it will be ignored.

**--mac macaddr** This is the same "-m macaddr".

**-d** Specifies that debugging output has to be produced on the Hercules control panel. This should normally be left unspecified.

**--debug** This is the same as "-d".

**guestip** This is an optional IP address of the Hercules (guest OS) side. This is only used to establish a point-to-point routing table entry on the driving system. If you use the --oat option do not specify an address here, it will be ignored.

### 5.9.5.4 OAT File Syntax

The syntax for the Address Translation file is as follows:

```
*******************************************************
* Dev   Mode  Port  Entry specific information...
*******************************************************
  0400  IP    00    PRI  172.021.003.032
  0402  IP    00    SEC  172.021.003.033
  0404  IP    00    NO   172.021.003.038
  0406  IP    01    NO   172.021.002.016
  040E  SNA   00
```

```
HWADD  00  02:00:FE:DF:00:42
HWADD  01  02:00:FE:DF:00:43
ROUTE  00  172.021.003.032  255.255.255.224
```

**Figure 2: OAT File Syntax**

**Dev** This is the base device address.

**Mode** This is the operation mode: IP or SNA. (Note: The SNA operation mode is currently not implemented.

**Port** This is the virtual (relative) adapter number.

**Entry** This applies only for IP Mode and specifies where a packet with an unknown IP address is forwarded. PRI is the primary default entry, SEC is the entry to use when the primary is not available and NO specifies that this is not a default entry.

**n.n.n.n** This specifies the home IP address.

When the operation mode is IP specify only the even (read) device number *dev.* The odd (write) address will be created automatically. Additionally two other statements can be included in the Address Translation file. These are the HWADD and ROUTE statements.

Use the HWADD statement to specify a hardware (MAC) address for a virtual adapter. The first parameter after the HWADD specifies the relative adpater for which the address is applied.

The ROUTE statement is included for convenience. It allows the hercifc program to create a network route for this specific virtual adapter. Please note that it is not necessary to include point-to-point routes for each IP address in the table. This is done automatically by the emulation module.

The read / write devices can be swapped by coding the odd address of the even-odd pair in the OAT. Up to 4 virtual (relative) adapters (00-03) are currently supported.

If no Address Translation file is specified, the emulation module will create the following:

- An ethernet adapter (port 0) for TCP/IP traffic only.

- Two device addresses (devnum and devnum+1).

### 5.9.5.5 Examples

**Example 1:**

Define LCS (LAN Channel Station emulation) adapters on device addresses 0440 and 0441. The name of the TUN/TAP special character device is “/dev/net/tun”. The IP address of the Hercules guest OS side is 192.168.200.2.

```
0440.2 LCS -n /dev/net/tun 192.168.200.2
```

## *5.9.6 VMNET (Channel-to-Channel link via SLIP/VMNET)*

### 5.9.6.1 Function

If the emulation mode is not specified on the configuration statement, it is assumed to be a point-to-point link to the driving system's TCP/IP stack using Willem Konynenberg's VMNET package.

This provides the same function as the CTCI mode of operation, except that it uses a virtual SLIP interface instead of the TUN/TAP driver.

## 5.10 FBA DASD Devices

### *5.10.1 Function*

This device statement is used to define a FBA DASD device to the Hercules configuration. The argument specifies the name of a file which contains the FBA DASD image or the INET address of a Hercules shared device server.

The file consists of fixed length 512-byte records each of which represents one physical block of the emulated disk. To allow access to a minidisk within a full-pack FBA DASD image file two additional arguments (origin and numblks) may be specified after the filename.

### *5.10.2 Syntax*

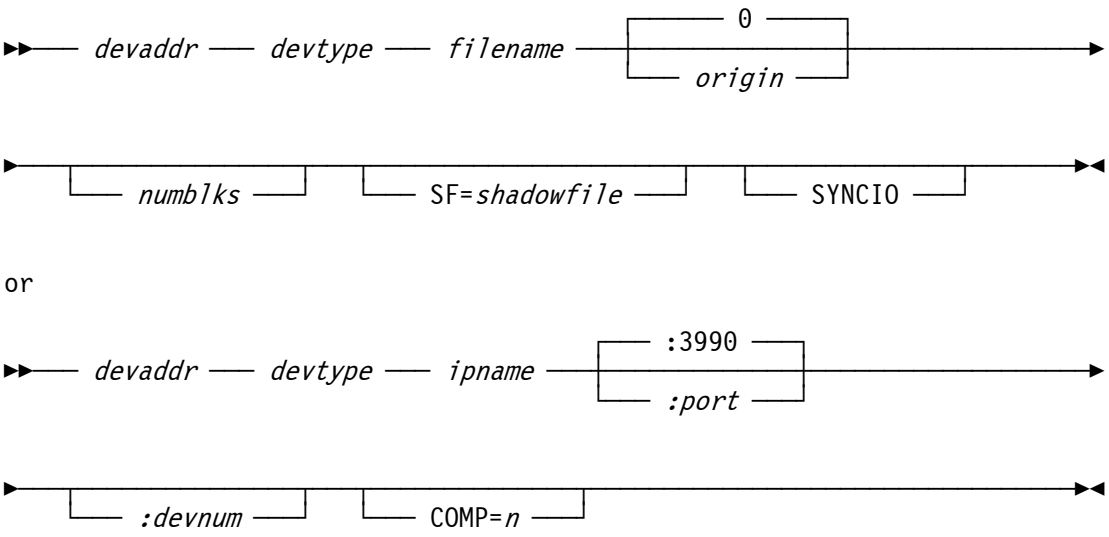
**Descriptive**

*devaddr devtype filename* [*origin* | 0] [*numblks*] [sf=*shadowfile*] [SYNCIO]

or

*devaddr devtype ipname* [*:port* | :3990] [*:devnum*] [COMP=*n*]

**Diagram**




### *5.10.3 Parameter*

**devaddr** This is the device address.

**devtype** This is the device type. Valid device types are 3310, 3370, 9332, 9335, 9336 and 0671.

**filename** This specifies the filename (and path) of the file containing the DASD image.

**origin** This defines the relative block number within the DASD image file at which the minidisk begins. The number must be less than the number of blocks in the file. The default origin is zero.

**numblks** Specifies the number of 512-byte blocks in the minidisk. This number must not exceed the number of blocks in the file minus the origin. If omitted or specified as an asterisk ("*"), then the minidisk continues to the end of the DASD image file.

**shadowfile** The handling of shadow files for FBA devices is identical as that for CCKD devices. Please refer to the following CCKD section for information regarding the use of the SF=shadow file option.

**SYNCIO** SYNCIO enables possible synchronous I/O and is explained in detail in the following CKD DASD section.

**ipname** This is the host name or IP adress of the system, where the Hercules shared device server is running.

**port** This is the port number the shared device server is listening on. If omitted the default is 3990.

**devnum** Devnum specifies the device number on the shared device server. If omitted the default is the current device number on the local system.

**COMP=n** This keyword requests that the data has to be transferred compressed between the client and the server. The argument *n* specifies the compression level (1-9). A value closer to 1 means less compression but also less processor time to perform the compression. A value closer to 9 means the data is compressed more but also more processor time is required to compress the data.

## *5.10.4 Examples*

**Example 1:**

Define a 3310 FBA DASD device on device address 0120. The name and path of the file containing the DASD image is "D:/DASD/TST001.FBA".

```
0120 3310 D:/DASD/TST001.FBA
```

**Example 2:**

Define a 3370 FBA DASD device on device address 0120. The DASD device is attached to a shared device server which is running on a machine with IP address 192.168.200.1 and listening on port 3990. The device address of the DASD on the shared device server is 0121.

```
0120 3370 192.168.200.1:3990:0121
```

## 5.11 CKD DASD Devices

### *5.11.1 Function*

This device statement is used to define a CKD DASD device to the Hercules configuration. The argument specifies the name of a file that contains the CKD DASD image or the INET address of a Hercules shared device server.

The file consists of a 512-byte device header record followed by fixed length track images. The length of each track image depends on the emulated device type and is always rounded up to the next multiple of 512 bytes.

Volumes larger than 2 GB (for example the 3390, model 3) can be supported by spreading the data across more than one file. Each file contains a whole number of cylinders. The first file (which contains cylinders 0-2518 in the case of a 3390) usually has "_1" as the last two characters of its name. The ckddasd driver allocates the remaining files by replacing the last characters of the file name by the characters "_2", "_3" and so on.

If your operating system supports large file sizes (or 64-bit offsets) then volumes larger than 2GB can be kept in a single file. The two character suffix is not used in this case.

Alternatively the argument may specify the name of a file containg a compressed CKD DASD image (CCKD DASD files). The CKD driver will automatically detect whether the file contains a regular CKD image or a compressed CKD image.

### *5.11.2 Syntax*

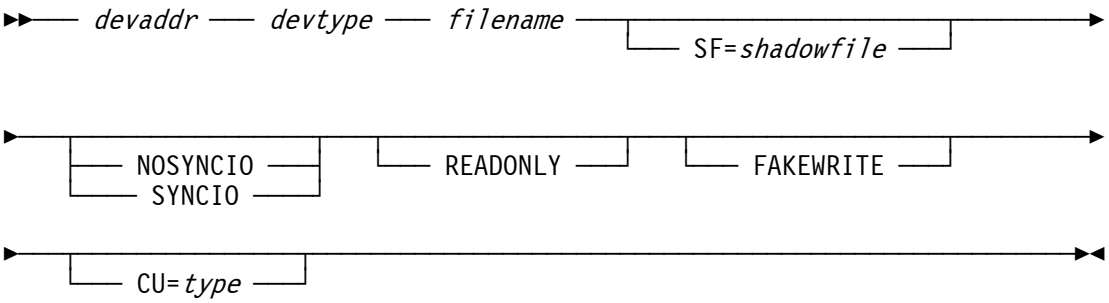
**Descriptive**

*devaddr devtype filename* [sf=*shadowfile*] [NOSYNCIO | SYNCIO] [READONLY]
[FAKEWRITE] [CU=*type*]

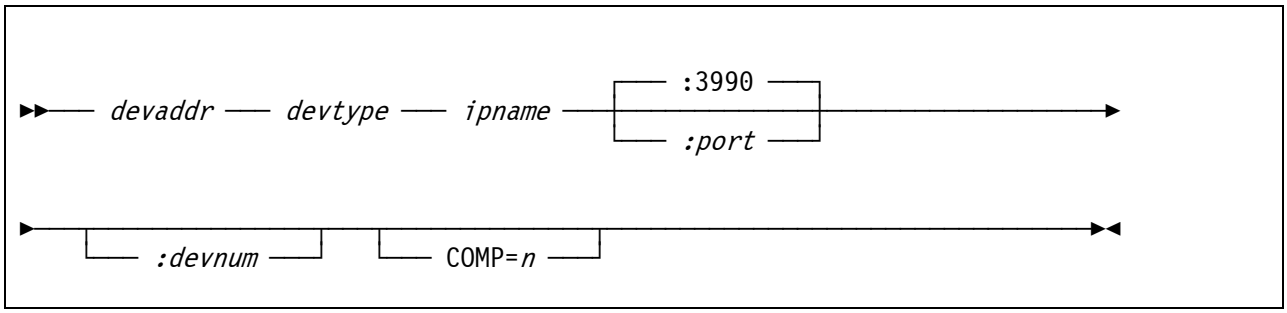
or

*devaddr devtype ipname* [*:port* | :3990] [*:devnum*] [COMP=*n*]

**Diagram**




or




## *5.11.3 Parameter*

**devaddr** This is the device address.

**devtype** This is the device type. Valid device types are 2311, 2314, 3330, 3350, 3375, 3380, 3390 and 9345.

**filename** This specifies the filename (and path) of the file containing the DASD image.

**shadowfile** A shadow file contains all the changes made to the emulated DASD device since it was created, maintained until the next shadow file is created. The moment of the shadow file's creation can be thought of a snapshot of the current emulated DASD at that time, as if the shadow file is later removed the emulated DASD reverts back to the state it was when the snapshot was taken.

Using shadow files, the base files can be kept on a read-only device like CDROM or can be defined as read-only thus ensuring that these files can never be corrupted.

The shadow file does not have to actually exist when it is defined in the configuration file. The *shadow* operand of the sf= parameter is simply a filename template that will be used to name the shadow file whenever one is be created. Shadow files are created using the sf+ xxxx or sf+ * commands.

The shadow file name must have a position where the shadow file number will be set. This is either the character preceeding the last period after the slash or the last character if there is no period. For example: *sf=shadows/linux1_*.dsk*.

Hercules console commands are provided to add a new shadow file, remove the current shadow file (with or without backward merge), compress the current shadow file and display the shadow file status and statistics.

Please note, that the "sf" parameter has to be coded in lowercase letters otherwise an error message is presented ("HHCDA003E parameter 2 is invalid: SF=…" ).

Details on how to work with shadow files can be found in chapter 7.94 ff.

**[NO]SYNCIO** SYNCIO enables possible "synchronous" I/O. This is a DASD I/O feature wherein guest I/O requests are completed "synchronously" during the actual emulated execution of the SIO / SSCH (START IO / START SUBCHANNEL) instructions rather than being deferred and executed asynchronously in a separate device I/O thread.

Only I/O which are known to be able to be completed without actually needing to perform any actual host I/O are completed synchronously (e.g. whenever the data being requested is found already be in the cache). If the requested data is not found in the cache then actual host I/O will need to be done and the request is passed to a device I/O thread to be completed asynchronously instead.

SYNCIO is the default for CKD devices. SYNCIO statistics may be displayed via the Hercules SYNCIO console command. SYNCIO may be abbreviated as *SYIO.*

Note: If you plan on using SYNCIO with Linux/390 and/or zLinux you might also want to take a look at the IODELAY configuration statement as well.

**READONLY** READONLY returns “write inhibited” sense when a write is attempted. Note that not all the sense bits may be set absolutely correctly.

**FAKEWRITE** FAKEWRITE is a workaround for the READONLY sense problem mentioned above. In these cases the disk is not intended to be updated (e.g. MVS updates the DSCB last referenced field for a readonly file) and any writes appear to be successful even though nothing is actually written.

**type** Specifies the type of control unit to which this device is attached. The use of this parameter does not necessarily imply that all functions of the specified control unit are emulated. The purpose is to force a particular control unit type to be indicated in the data returned by SENSE ID and similar CCW's.

Normally the default value is appropriate and this parameter need not to be specified. Additional to the table below the following values may be specified: 3990-3 and 3990-6.

The default value depends on the device type:

| Device Type | Default CU Type |
|---|---|
| 2311 | 2841 |
| 2314 | 2314 |
| 3330, 3340, 3350, 3375, 3380 | 3880 |
| 3390 | 3990 |
| 9345 | 9343 |

**Table 6: Default CU Types**

**ipname** This is the host name or IP adress of the system where the Hercules shared device server is running.

**port** This is the port number the shared device server is listening on. If omitted the default is 3990.

**devnum** Devnum specifies the device number on the shared device server. If omitted the default is the current device number on the local system.

**COMP=n** This keyword requests that the data has to be transferred compressed between the client and the server. The argument *n* specifies the compression level (1-9). A value closer to 1 means less compression but also less processor time to perform the compression. A value closer to 9 means the data is compressed more but also more processor time is required to compress the data.

Two other options *LAZYWRITE* and *FULLTRACKIO* have been deprecated. They are still accepted to support compatibility with older configuration files but do nothing. It is strongly recommended to remove these statements from the configuration file.

### *5.11.4 Examples*

**Example 1:**

Define a 3390 CKD DASD device on device address 0A8C. The name and path of the file containing the DASD image is "D:/DASD/SARES1.CKD".

```
0A8C 3390 D:/DASD/SARES1.CKD
```

**Example 2:**

Define a 3390 CKD DASD device on device address 0AA2. The name and path of the file containing the DASD image is "D:/DASD/SARES1.CKD". Synchronous I/O has to be enabled and the device has to return the "write inhibited" sense when a write is attempted.

```
0AA2 3390 D:/DASD/SARES1.CKD SYNCIO READONLY
```

**Example 3:**

Define a 3390 CCKD DASD device on device address 0A00. The name and path of the file containing the DASD image is "D:/DASD/SARES1.CCKD". All changes to the DASD device have to go to the shadow file(s) "D:/SHADOW/SARES1_*.CCKD".

```
0A00 3390 D:/DASD/SARES1.CCKD SF=D:/SHADOW/SARES1_*.CCKD
```

**Example 4:**

Define a 3390 CKD DASD device on device address 0A8C. The DASD device is attached to a shared device server which is running on a machine with IP address 192.168.200.1 and listening on port 3990. The device address of the DASD on the shared device server is also 0A8C.

```
0A8C 3390 192.168.200.1:3990:0A8C
```

## 5.12 Communication Lines (Preliminary 2703 BSC Support)

### *5.12.1 Function*

This statement describes a BSC emulation line entry to either link 2 Hercules engines or a custom made program emulating a 2780, 3780 or 3x74, or a custom made program interfacing to a real BSC line.

The communication is emulated over a TCP connection. All bytes are transferred as-is (except for doubling DLE in transparent mode) just as over a real BSC link. Emulated EIA (DCD, DTR, CTS, etc.) or X.21/V.11 leads (C, T, etc.) are treated differently depending on the DIAL option selected.

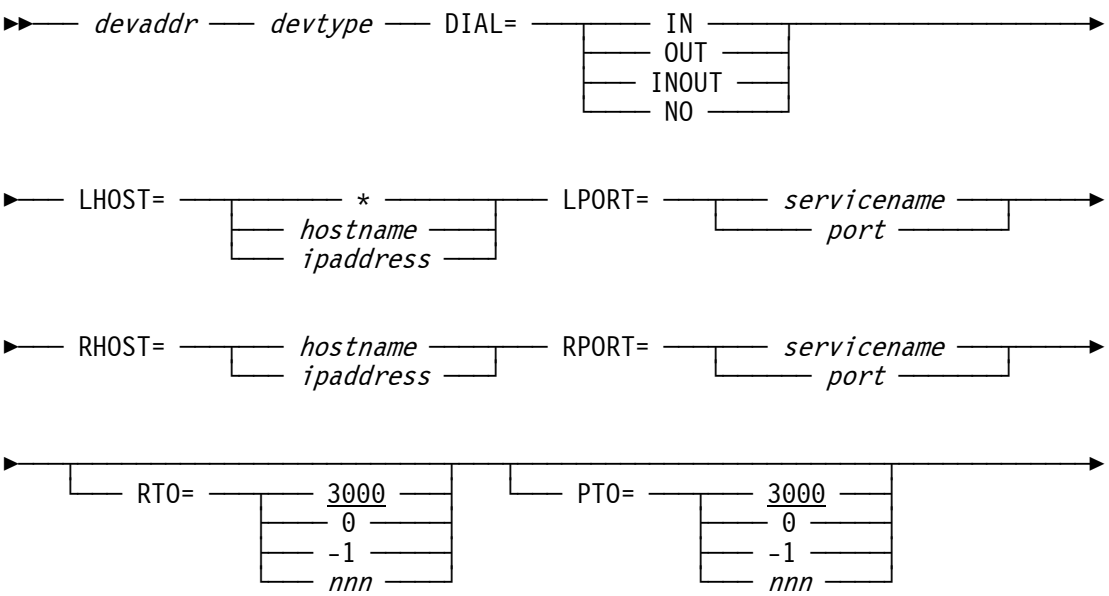
The line emulates a point-to-point BSC link. There is no point-to-multipoint handling.

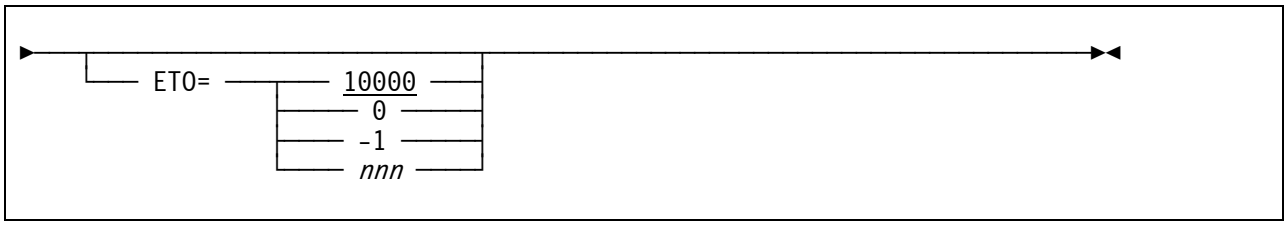
### *5.12.2 Syntax*

**Descriptive**

```
devaddr devtype DIAL={IN | OUT | INOUT | NO}
                LHOST={hostname | ipaddress | *} LPORT={servicename | port}
                RHOST={hostname | ipaddress } RPORT={servicename | port}
                [RTO={0 | -1 | nnn | 3000}]
                [PTO={0 | -1 | nnn | 3000}]
                [ETO={0 | -1 | nnn | 10000}]
```

**Diagram**






## 5.12.3 Parameter

**devaddr** This is the device address.

**devtype** This is the device type. A valid device type is 2703.

**DIAL** This specifies the call direction (if any). If DIAL=NO is specified the TCP outgoing connection is attempted as soon as an ENABLE CCW is executed. Also in this mode, an incoming connection will always be accepted. If DIAL=IN or INOUT is specified a TCP incoming call is accepted only if an ENABLE CCW is currently exectuting on the device. With DIAL=OUT the ENABLE CCW is rejected. When DIAL=IN or INOUT is coded a DIAL CCW allows the application to establish a TCP connection to a specific host. For other DIAL values the DIAL CCW is rejected.

**LHOST** Specifies which IP address to listen on. This also conditions the network interface from which incoming calls will be accepted. An asterisk (“*”) means all incoming TCP calls are accepted regardless of the destination IP address or call origin. A given IP address when DIAL=OUT is specified has no effect.

**LPORT** Specifies the TCP port for which to listen to incoming TCP calls. This value is mandatory for DIAL=IN, DIAL=INOUT or DIAL=NO. It is ignored for DIAL=OUT.

**RHOST** Specifies the remote host to which to direct a TCP connection on a DIAL=NO line when an ENABLE CCW is executed. This value is mandatory when DIAL=NO is specified, It is ignored for other DIAL values.

**RPORT** Specifies the remote port to which to direct a TCP connection on a DIAL=NO line when an ENABLE CCW is executed. This value is mandatory when DIAL=NO is specified, It is ignored for other DIAL values.

The following arguments are tuning options. In most cases using the default values will give the best result.

**RTO** Specifies the number of milliseconds before terminating a read on a timeout when no read termination control character is received. A value of zero means the read ends immediately. “-1” specifies that there is no timeout.

**PTO** Specifies the number of milliseconds before terminating a POLL on a timeout when no ACK or NACK sequence is received. A value of zero means the POLL ends immediately. “-1” specifies that there is no timeout.

**ETO** Specifies the number of milliseconds before terminating an ENABLE operation on a timeout. The timeout applies when DIAL=NO, DIAL=IN or DIAL=INOUT is specified, the outgoing TCP call fails (DIAL=NO) and there is no previously or currently established TCP connection for this line. When DIAL=NO is specified the timeout defaults to 10 seconds, for DIAL=IN or DIAL=INOUT, the timeout defaults to “-1”, meaning there is no timeout.

### *5.12.4 Examples*

**Example 1:**

Define 2703 BSC emulation line on device address 0023. The outgoing connection should be attempted as soon as an ENABLE CCW is expected, incoming connections should always be accepted (DIAL=NO). The IP address to listen on is 127.0.0.1 with port 3780. The remote port to which to direct a TCP connection is 3781.

```
0023 2703 DIAL=NO LHOST=LOCALHOST LPORT=3780 RPORT=3781
```

# 6. Hercules Console

## 6.1 Hercules Hardware Console

Various aspects of Hercules can be controlled with console commands. These commands are entered from the command line of the Hercules hardware management console (HMC).

The Hercules hardware management console window is the panel that is active by default after starting Hercules. An alternate semi-graphical "New Panel" display showing devices, registers etc. is also available. Pressing the escape key switches between the two panels.

The main Hercules screen contains a scrollable list of messages with a command input area and a system status line at the bottom of the screen. To scroll through the messages, use either the Page Up or Page Down keys, the Ctrl + Up Arrow or Ctrl + Down Arrow keys, or the Home or End and/or the Ctrl + Home or Ctrl + End keys.

Important messages are highlighted in a different color (usually red) and are prevented from being scrolled off the screen for two minutes. If Extended Cursor handling is available then important messages currently at the top of the screen can be removed early by moving the cursor to the line containing the message and then pressing enter.

Use the Insert key to switch between insert and overlay mode when typing in the command input area. Use the Home and End keys to move to the first or last character of the command you are typing, or the use the left/right arrow keys to move to a specific character. Use the Escape key to erase the input area.

When a command is prefixed with '-', the command will not be redisplayed at the console. This can be used in scripts and is also used internally when commands are to be invoked without being redisplayed at the console.

Pressing "Escape" when the command input area is already empty causes the screen to switch to the semi-graphical "New Panel" display mode, which shows the overall status of the system and devices.

When in the semi-graphical "New Panel" display mode there is no command input area. Instead, single character "hot keys" are used to issue some of the more common functions such as starting or stopping the CPU. The hot-keys are those which are highlighted. Pressing the '?' key displays brief help information on how to use the semi-graphical panel.

Sample windows are shown in the pictures on the following pages.

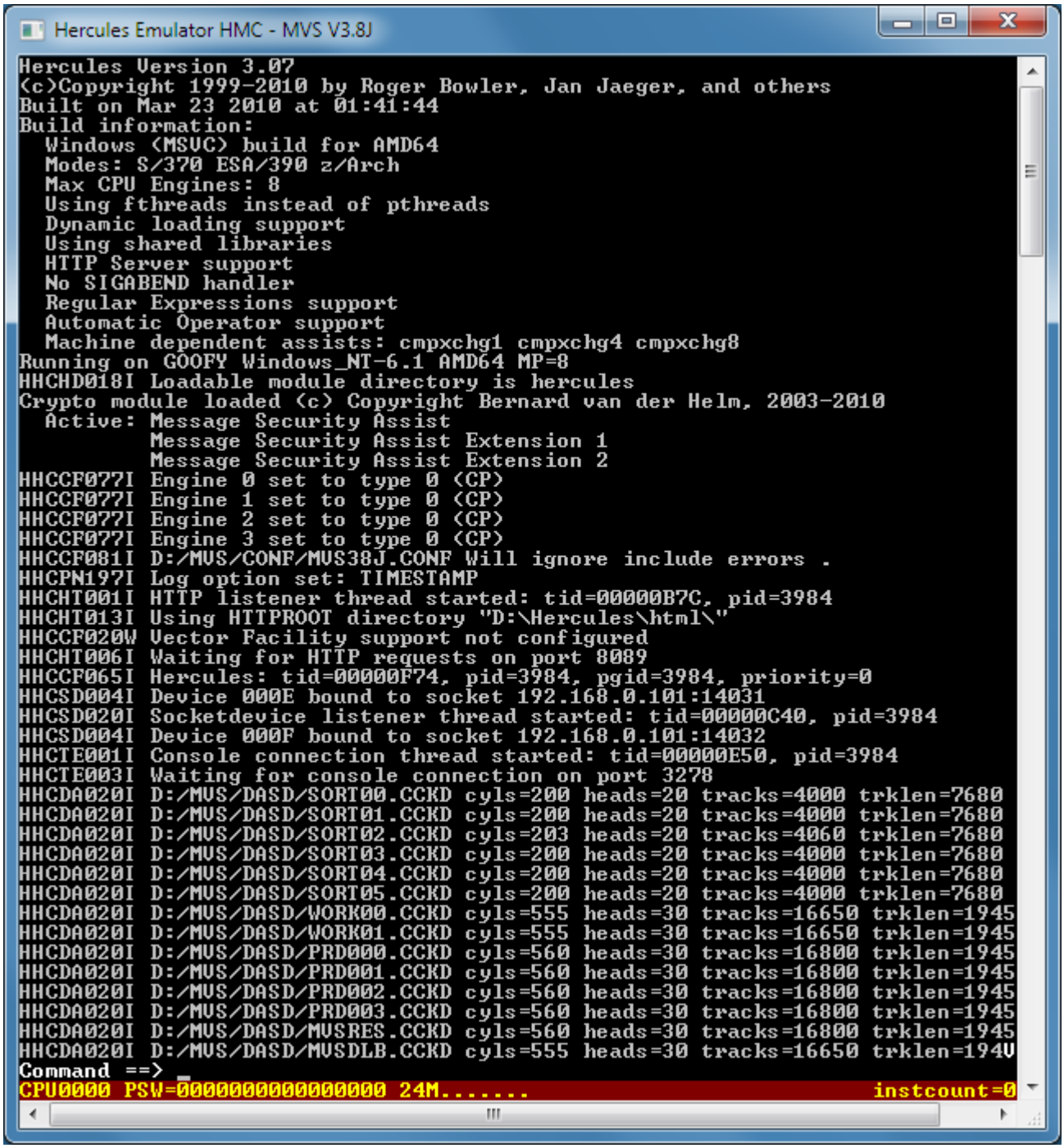



**Figure 3: Hercules Hardware Console**

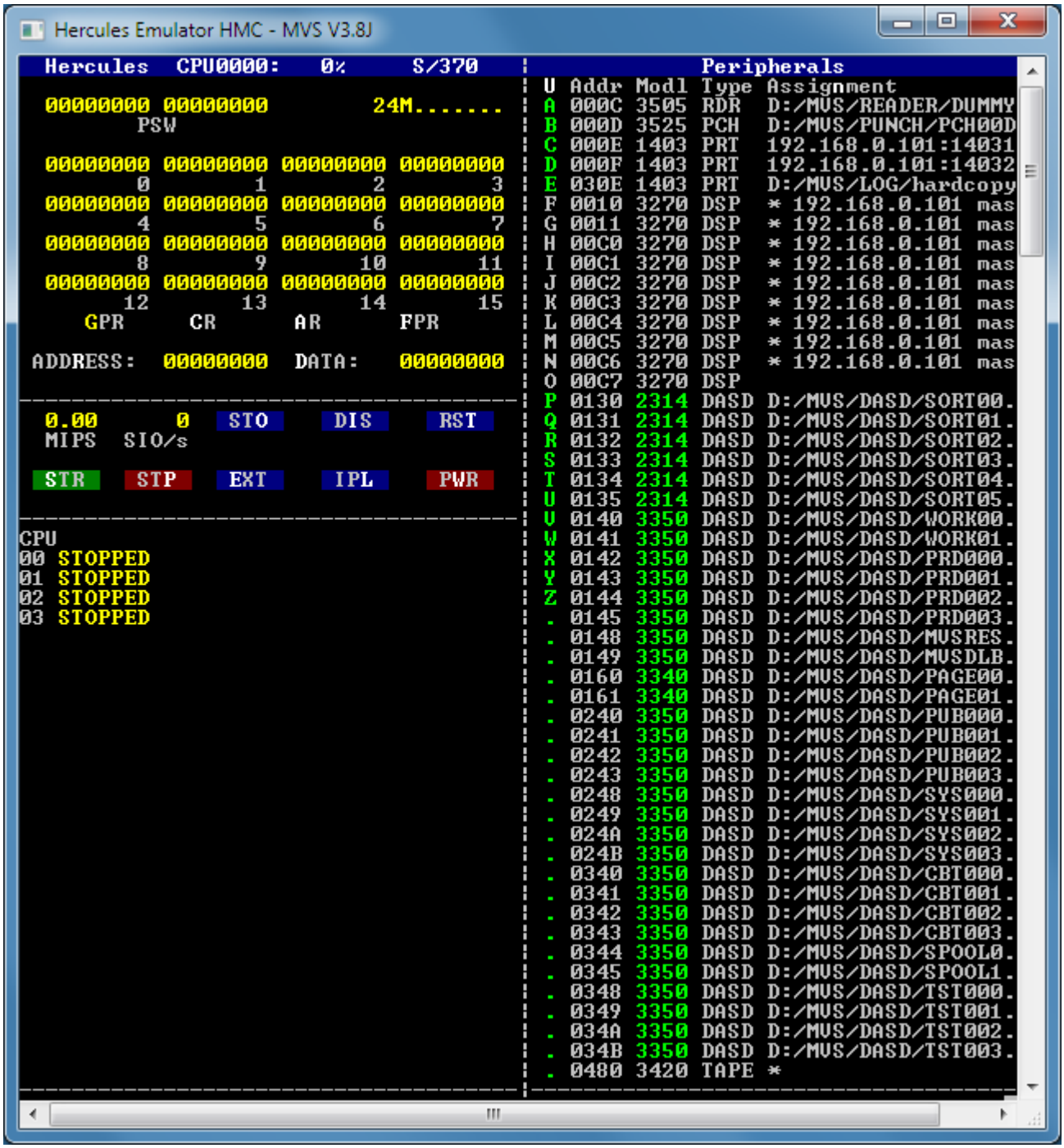



**Figure 4: Hercules device and status panel**

## 6.2 Web browser interface

It is not necessary to work directly on the PC where Hercules is running to enter the console commands. Instead of using the hardware management console on the Hercules server itself it is possible to work from a browser interface in a web browser.

The requirement to be able to work with the web browser interface is a running Hercules HTTP server. The HTTP server is started through the HTTPPORT and HTTPROOT configuration statements. For details on configuring the HTTP server please see section "System Parameter Descriptions".

Once the HTTP server is started, the interface can be used from a web browser on any machine, that has a network connection to the Hercules server. With this interface you can operate Hercules nearly in the same way as you would do locally on the hardware management console.

The following picture shows the main panel of the web browser interface.

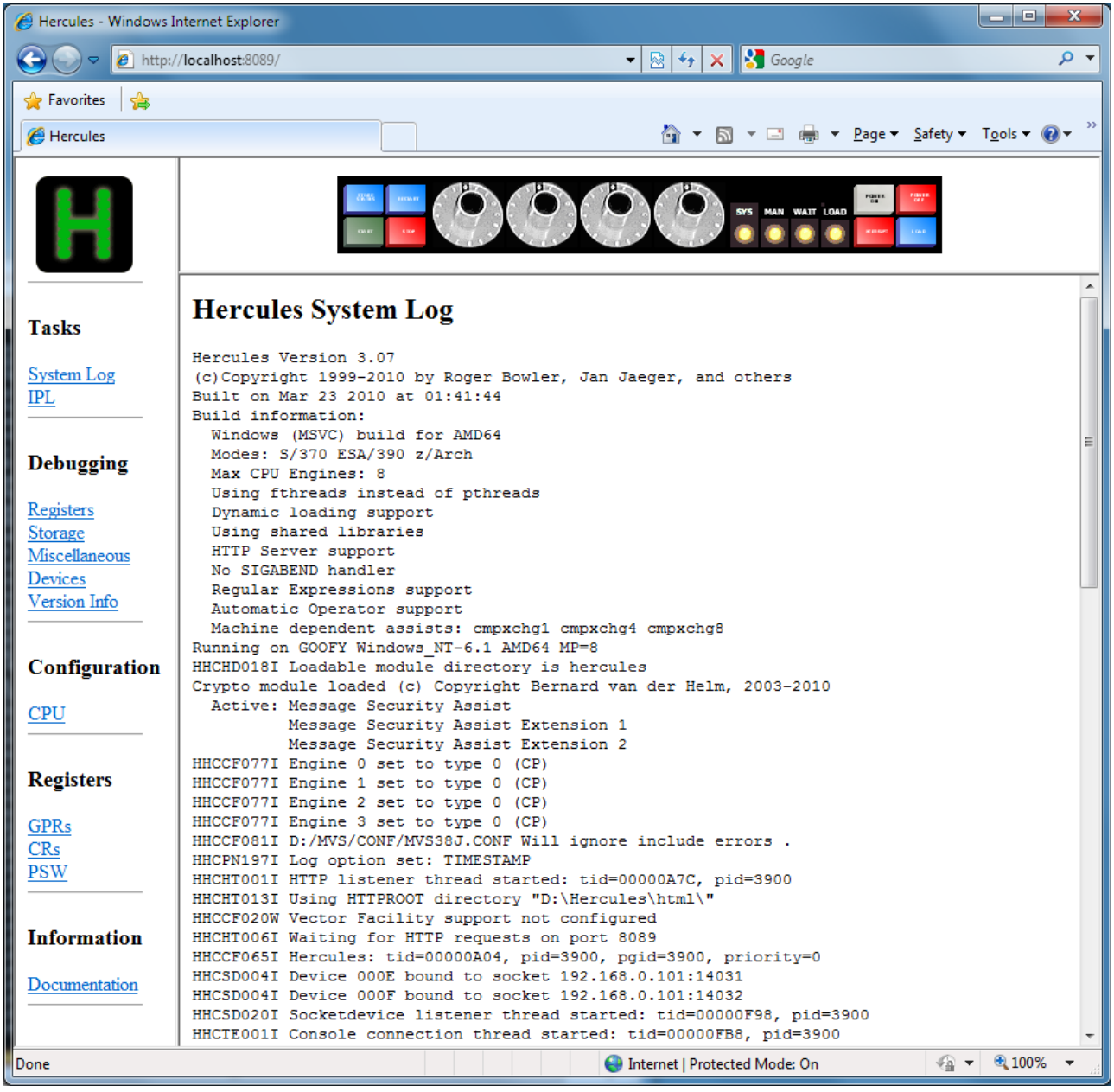



**Figure 5: Hercules web browser interface**

## 6.3 Using the keyboard

The usage of the keyboard in the Hercules hardware management console is distinguished between normal cursor handling and extended cursor handling.

### 6.3.1 Normal cursor handling

The normal cursor handling is available on all platforms (Windows and Unix).

| Normal cursor handling | |
|---|---|
| **Key** | **Action** |
| Esc | Erases the contents of the command input area. If the command input area is already empty, switches to semi-graphical New Panel. |
| Del | Deletes the character at the cursor position. |
| Backspace | Erases the previous character. |
| Insert | Toggles between insert mode and overlay mode. |
| Tab | Attempts to complete the partial file name at the cursor position in the command input area. If more than one possible file exists, a list of matching file names is displayed. |
| Home | Moves the cursor to the start of the input in the command input area. If the command input area is empty, scrolls the message area to the top. |
| End | Moves the cursor to the start of the input in the command input area. If the command input area is empty, scrolls the message area to the bottom. |
| Page Up | Scrolls the message area up one screen. |
| Page Down | Scrolls the message area down one screen. |
| Up arrow | Recalls the previous command into the input area. |
| Down arrow | Recalls the next command into the input area. |
| Right arrow | Moves cursor to the next character of the input area. |
| Left arrow | Moves cursor to the previous character of the input area. |
| Ctrl + Up arrow | Scrolls the message area up one line. |
| Ctrl + Down arrow | Scrolls the message area down one line. |
| Ctrl + Home | Scrolls the message area to the top. |
| Ctrl + End | Scrolls the message area to the bottom. |

**Table 7: Normal cursor handling**

### 6.3.2 Extended cursor handling

The following additional keyboard functions are effective when the Hercules Extended Cursor Handling feature (OPTION_EXTCURS) is activated at compile time. At present, this feature is activated on the Windows platform only.

| Extended cursor handling | |
|---|---|
| **Key** | **Action** |
| Alt + Up arrow | Moves cursor up one row. |
| Alt + Down arrow | Moves cursor down one row. |
| Alt + Right arrow | Moves cursor right one column. |
| Alt + Left arrow | Moves cursor left one column. |
| Tab | If the cursor is outside the command input area, moves cursor to the start of the input in the command input area. Otherwise behaves like as described in the previous table. |
| Home | If the cursor is outside the command input area, moves cursor to the start of the input in the command input area. Otherwise behaves like as described in the previous table. |
| End | If the cursor is outside the command input area, moves cursor to the end of the input in the command input area. Otherwise behaves like as described in the previous table. |

**Table 8: Extended cursor handling**

## 6.4 Log formats

The appearance of the log in terms of prefixing the messages is dependent on the used environment. The most flexible ways of prefixing the messages is given when running Hercules using the WinGUI. Running Hercules with the native console gives only limited options. The following table show an overview of the log options, depending on the Hercules version used.

| Message Prefix | Native Hercules Console | Hercules with WinGUI |
|---|---|---|
| Date | No | Yes |
| Time | Yes | Yes |
| Process ID | No | Yes |
| Prefix(es) visible on console | No | Yes |
| Prefix(es) visible on hardcopy log | Yes | Yes |

**Table 9: Message prefix overview**

Most of the console command examples in this book are made using the WinGUI as described below. In some examples the log options have been reduced to fit one log line into the example boxes in the manual.

## 6.4.1 Hercules Console

Using Hercules under Linux, Mac OS or Windows (without WinGUI) lets you prefix the consol messages with a timestamp. The difference between these options is shown in the next figure. Switching the option can be done with the logopt console command or can be set at startup time through the logopt system parameter.

Please note that the console itself does not show the timestamp, independent of the logopt settings. Only the hardcopy log does show the timestamp.

```
.
.
.
logopt notimestamp
HHCPN197I Log option set: NOTIMESTAMP
gpr
CPU0000: R0=0000000000000000 R1=0000000000000000
CPU0000: R2=0000000000000000 R3=0000000000000000
CPU0000: R4=0000000000000000 R5=0000000000000000
CPU0000: R6=0000000000000000 R7=0000000000000000
CPU0000: R8=0000000000000000 R9=0000000000000000
CPU0000: RA=0000000000000000 RB=0000000000000000
CPU0000: RC=0000000000000000 RD=0000000000000000
CPU0000: RE=0000000000000000 RF=0000000000000000
23:01:32 logopt timestamp
23:01:32 HHCPN197I Log option set: TIMESTAMP
23:01:36 gpr
23:01:36 CPU0000: R0=0000000000000000 R1=0000000000000000
23:01:36 CPU0000: R2=0000000000000000 R3=0000000000000000
23:01:36 CPU0000: R4=0000000000000000 R5=0000000000000000
23:01:36 CPU0000: R6=0000000000000000 R7=0000000000000000
23:01:36 CPU0000: R8=0000000000000000 R9=0000000000000000
23:01:36 CPU0000: RA=0000000000000000 RB=0000000000000000
23:01:36 CPU0000: RC=0000000000000000 RD=0000000000000000
23:01:36 CPU0000: RE=0000000000000000 RF=0000000000000000.
.
.
```

**Figure 6: Message prefixing using the native Hercules console**

## 6.4.2 Hercules using the WinGUI

Using the WinGUI with the Windows version of Hercules gives you the possibility to prefix all messages with the current date, the current time and the process ID. These prefixing options can be selected individually in the “Format” tab of the "Advanced Logging Options", where they can be separately turned on or off as showed in the next figure.

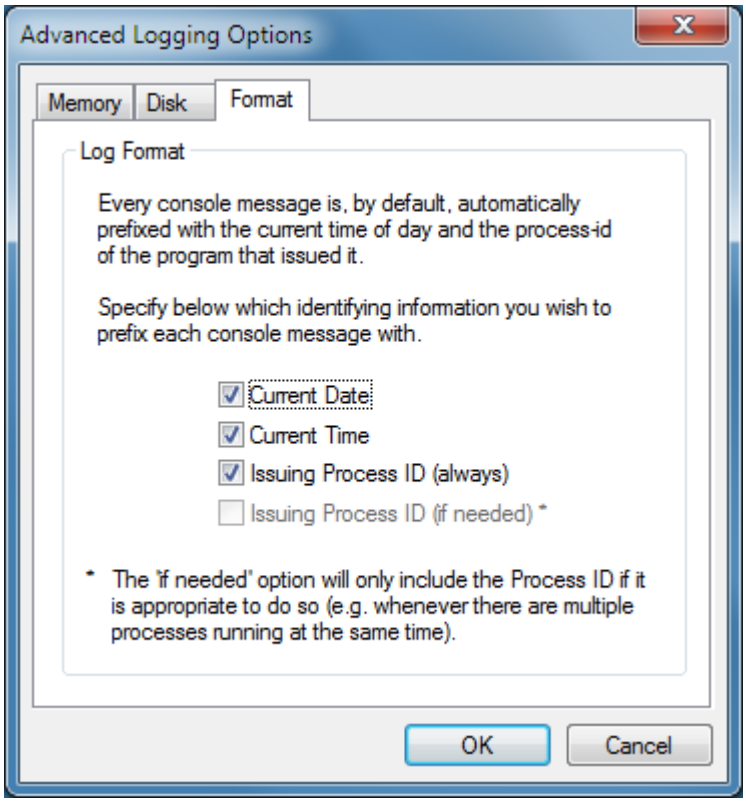



**Figure 7: WinGUI Advanced Logging Options**

The logopt system parameter or console command has no effect when using the WinGUI. Prefixing the messages is only influenced over the GUI settings. The following figure shows the log messages using all prefixing options (date, time and process ID) as specified above.

```
.
.
.
2010/06/18 19:12:18.071 000008E4 HHC01416I Build information:
2010/06/18 19:12:18.071 000008E4 HHC01417I Windows (MSVC) build for AMD64
2010/06/18 19:12:18.071 000008E4 HHC01417I Modes: S/370 ESA/390 z/Arch
2010/06/18 19:12:18.071 000008E4 HHC01417I Max CPU Engines: 8
2010/06/18 19:12:18.071 000008E4 HHC01417I Using fthreads Threading Model
2010/06/18 19:12:18.072 000008E4 HHC01417I Using FishIO
2010/06/18 19:12:18.072 000008E4 HHC01417I Dynamic loading support
2010/06/18 19:12:18.072 000008E4 HHC01417I Using shared libraries
2010/06/18 19:12:18.072 000008E4 HHC01417I HTTP Server support
2010/06/18 19:12:18.072 000008E4 HHC01417I No SIGABEND handler
2010/06/18 19:12:18.072 000008E4 HHC01417I Regular Expressions support
2010/06/18 19:12:18.072 000008E4 HHC01417I Automatic Operator support
.
.
.
```

**Figure 8: Message prefixing using the console of the Hercules WinGUI**

## 6.5 Hercules Console Commands (sorted alphabetically)

The following table shows an overview of the valid Hercules console commands that can be entered in the control panel (Hercules console display). In this table the commands are sorted alphabetically rather than grouped by functionality.

| Command | Description |
|---|---|
| !message | SCP priority message |
| # | Log comment to syslog |
| * | Log comment to syslog |
| .reply | SCP command |
| ? | List all commands / command specific help (alias for help) |
| aea | Display AEA (absolue-effective-address) tables |
| aia | Display AIA (absolute-instruction-address) tables |
| ar | Display access registers |
| archmode | Set architecture mode |
| attach | Configure device |
| auto_scsi_mount | Automatic SCSI tape mounts (synonym for SCSIMOUNT) |
| automount | Show or update allowable tape automount directories |
| b | Set breakpoint (synonym for B+) |
| b+ | Set breakpoint (synonym for B) |
| b- | Delete breakpoint |
| cache | Cache command |
| cckd | CCKD command |
| cd | Change directory |
| cf | Configure CPU online or offline |
| cfall | Configure all CPU’s online or offline |
| clocks | Display TOD clock and CPU timer |
| cmdtgt | Specify the command target |
| conkpalv | Display / alter console TCP/IP keep-alive settings |

| Command | Description |
|---|---|
| cpu | Define target CPU for console display and commands |
| cr | Display or alter control registers |
| cscript | Cancel a running script thread |
| ctc | Enable / disable CTC debugging |
| define | Rename device |
| defsym | Define symbol |
| detach | Remove device |
| devinit | Reinitialize device |
| devlist | List device or all devices |
| devtmax | Display or set max device threads |
| ds | Display subchannel |
| ecpsvm | ECPS:VM commands |
| exit | Synonym for “quit” |
| ext | Generate external interrupt |
| f{+/-} addr | Mark frames usable / unusable |
| fpc | Display floating point control registers |
| fpr | Display floating point register |
| g | Turn off instruction stepping and start CPU |
| gpr | Display or alter general purpose registers |
| hao | Hercules Automatic Operator (HAO) |
| help | List all commands / command specific help |
| herc | Send Hercules command |
| herclogo | Read a new logo file |
| hst | History of commands |
| i | Generate I/O attention interrupt for device |
| iodelay | Display or set I/O delay value |

| Command | Description |
|---|---|
| ipending | Display pending interrupts |
| ipl | IPL normal from device xxxx |
| iplc | IPL clear from device xxxx |
| k | Display CCKD internal trace |
| ldmod | Load a module |
| loadcore | Load a core image file |
| loadparm | Set IPL parameter |
| loadtext | Load a text deck file |
| log | Direct log output |
| logopt | Change log options |
| lparname | Display or set LPAR name |
| lparnum | Display or set LPAR identification number |
| lsdep | List module dependencies |
| lsmod | List dynamic modules |
| maxrates | Display maximum observed MIPS/SIO rate or define new reporting interval |
| message | Display message on console like VM |
| mounted_tape_reinit | Control tape initialization |
| msg | Display message on console like VM |
| msghld | Display or set timeout value of held messages |
| msgnoh | Display message on console like VM, but without header |
| ostailor | Specify intended operating system |
| panrate | Display or set console refresh rate |
| pgmtrace | Trace program interrupts |
| pr | Display prefix register |
| pscp | Send system control program priority message |
| psw | Display or alter program status word |

| Command | Description |
|---|---|
| ptt | Set / display pthread trace |
| pwd | Print working directory |
| qd | Query DASD |
| quiet | Toggle automatic refresh of console display data |
| quit | Terminate the emulator |
| r | Display or alter real storage |
| restart | Generate restart interrupt |
| resume | Resume Hercules |
| rmmod | Delete a module |
| s | Instruction stepping |
| s+ | Instruction stepping on |
| s- | Instruction stepping off |
| s? | Instruction stepping query |
| s{+/-} dev | Turn CCW stepping on / off |
| savecore | Save a core image file |
| sclproot | Set or display SCLP base directory |
| scp | Send system control program command |
| script | Run a sequence of console commands contained in a file |
| scsimount | Automatic SCSI tape mounts (synonym for AUTO_SCSI_MOUNT) |
| sf+ | Create a new shadow file |
| sf- | Remove a shadow file |
| sfc | Compress a shadow file |
| sfd | Display shadow file statistics |
| sfk | Perform a chkdsk on the active shadow file |
| sh | Shell command |
| shrd | SHRD command |

| Command | Description |
|---|---|
| sizeof | Display size of structures |
| ssd | Signal Shutdown |
| start | Start CPU or printer device |
| startall | Start all CPU's |
| stop | Stop CPU or printer device |
| stopall | Stop all CPU's |
| store | Store CPU status at absolute zero |
| suspend | Suspend Hercules |
| syncio | Display syncio device statistics |
| sysclear | Issue SYSTEM CLEAR RESET manual operation |
| sysreset | Issue SYSTEM RESET manual operation |
| t | Instruction trace |
| t+ | Instruction trace on |
| t- | Instruction trace off |
| t? | Instruction trace query |
| t{+/-} CKD | Turn CKD_KEY tracing on / off |
| t{+/-} dev | Turn CCW tracing on / off |
| timerint | Display or set timers update interval |
| tlb | Display TLB tables |
| toddrag | Display or set TOD clock drag factor |
| traceopt | Instruction trace display option |
| tt32 | Control / query CTCI-WIN functionality |
| u | Disassemble storage |
| uptime | Display Hercules Emulator uptime |
| v | Display or alter virtual storage |
| version | Display version information |

**Table 10: Hercules Console Commands (sorted alphabetically)**

## 6.6 Hercules Console Commands (grouped by functionality)

The next table shows an overview of the valid Hercules console commands that can be entered in the control panel (Hercules console display). In this table the commands are grouped by functionality.

| Command | Description |
|---|---|
| help | List all commands / command specific help |
| ? | List all commands / command specific help (alias for help) |
| | |
| # | Log comment to syslog |
| * | Log comment to syslog |
| | |
| message | Display message on console like VM |
| msg | Display message on console like VM |
| msgnoh | Display message on console like VM, but without header |
| | |
| quiet | Toggle automatic refresh of console display data |
| | |
| hst | History of commands |
| hao | Hercules Automatic Operator (HAO) |
| log | Direct log output |
| logopt | Change log options |
| uptime | Display Hercules Emulator uptime |
| version | Display version information |
| | |
| quit | Terminate the emulator |
| exit | Synonym for “quit” |
| | |
| cpu | Define target CPU for console display and commands |
| | |
| start | Start CPU or printer device |
| stop | Stop CPU or printer device |
| | |
| startall | Start all CPU’s |
| stopall | Stop all CPU’s |
| | |

| Command | Description |
|---|---|
| cf | Configure CPU online or offline |
| cfall | Configure all CPU's online or offline |
| | |
| .reply | SCP command |
| !message | SCP priority message |
| ssd | Signal Shutdown |
| | |
| ptt | Set / display pthread trace |
| | |
| i | Generate I/O attention interrupt for device |
| ext | Generate external interrupt |
| restart | Generate restart interrupt |
| archmode | Set architecture mode |
| loadparm | Set IPL parameter |
| lparname | Display or set LPAR name |
| lparnum | Display or set LPAR identification number |
| | |
| ipl | IPL normal from device xxxx |
| iplc | IPL clear from device xxxx |
| sysreset | Issue SYSTEM RESET manual operation |
| sysclear | Issue SYSTEM CLEAR RESET manual operation |
| store | Store CPU status at absolute zero |
| | |
| sclproot | Set or display SCLP base directory |
| | |
| psw | Display or alter program status word |
| gpr | Display or alter general purpose registers |
| fpr | Display floating point register |
| fpc | Display floating point control registers |
| cr | Display or alter control registers |
| ar | Display access registers |

| Command | Description |
|---|---|
| pr | Display prefix register |
| timerint | Display or set timers update interval |
| clocks | Display TOD clock and CPU timer |
| ipending | Display pending interrupts |
| ds | Display subchannel |
| r | Display or alter real storage |
| v | Display or alter virtual storage |
| u | Disassemble storage |
| devtmax | Display or set max device threads |
| k | Display CCKD internal trace |
| | |
| attach | Configure device |
| detach | Remove device |
| define | Rename device |
| devinit | Reinitialize device |
| devlist | List device or all devices |
| | |
| qd | Query DASD |
| | |
| mounted_tape_reinit | Control tape initialization |
| auto_scsi_mount | Automatic SCSI tape mounts (synonym for SCSIMOUNT) |
| scsimount | Automatic SCSI tape mounts (sysnonym for AUTO_SCSI_MOUNT) |
| automount | Show or update allowable tape automount directories |
| | |
| cd | Change directory |
| pwd | Print working directory |
| sh | Shell command |
| | |
| cache | Cache command |
| cckd | CCKD command |

| Command | Description |
|---|---|
| shrd | SHRD command |
| conkpalv | Display / alter console TCP/IP keep-alive settings |
| | |
| t | Instruction trace |
| t+ | Instruction trace on |
| t- | Instruction trace off |
| t? | Instruction trace query |
| s | Instruction stepping |
| s+ | Instruction stepping on |
| s- | Instruction stepping off |
| s? | Instruction stepping query |
| b | Set breakpoint (synonym for B+) |
| b+ | Set breakpoint (synonym for B) |
| b- | Delete breakpoint |
| g | Turn off instruction stepping and start CPU |
| | |
| ostailor | Specify intended operating system |
| pgmtrace | Trace program interrupts |
| savecore | Save a core image file |
| loadcore | Load a core image file |
| loadtext | Load a text deck file |
| | |
| ldmod | Load a module |
| rmmod | Delete a module |
| lsmod | List dynamic modules |
| lsdep | List module dependencies |
| | |
| iodelay | Display or set I/O delay value |
| ctc | Enable / disable CTC debugging |

| Command | Description |
|---|---|
| tt32 | Control / query CTCI-WIN functionality |
| toddrag | Display or set TOD clock drag factor |
| panrate | Display or set console refresh rate |
| msghld | Display or set timeout value of held messages |
| syncio | Display syncio device statistics |
| maxrates | Display maximum observed MIPS/SIO rate or define new reporting interval |
| | |
| defsym | Define symbol |
| script | Run a sequence of console commands contained in a file |
| cscript | Cancel a running script thread |
| | |
| ecpsvm | ECPS:VM commands |
| | |
| aea | Display AEA (absolue-effective-address) tables |
| aia | Display AIA (absolute-instruction-address) tables |
| tlb | Display TLB tables |
| | |
| sizeof | Display size of structures |
| | |
| suspend | Suspend Hercules |
| resume | Resume Hercules |
| | |
| herclogo | Read a new logo file |
| | |
| traceopt | Instruction trace display option |
| | |
| cmdtgt | Specify the command target |
| herc | Send Hercules command |
| scp | Send system control program command |
| pscp | Send system control program priority message |
| | |
| sf+ | Create a new shadow file |
| sf- | Remove a shadow file |

| Command | Description |
|---|---|
| sfc | Compress a shadow file |
| sfd | Display shadow file statistics |
| sfk | Perform a chkdsk on the active shadow file |
| | |
| t{+/-} dev | Turn CCW tracing on / off |
| s{+/-} dev | Turn CCW stepping on / off |
| t{+/-} CKD | Turn CKD_KEY tracing on / off |
| f{+/-} addr | Mark frames usable / unusable |

**Table 11: Hercules Console Commands (grouped by functionality)**

# 7. Console Command Descriptions

## 7.1 !message (SCP priority message)

### *7.1.1 Function*

The “!message” (SCP) command is used to enter a system control program (i.e. guest operating system) priority command on the hercules console. The command to be issued has to be prefixed with an exclamation point '!'.

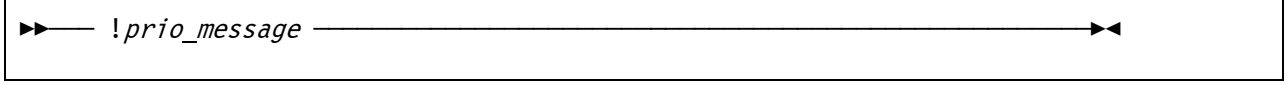
This command is similar to the “.reply” command. It is up to the operating system to differentiate between them.

### *7.1.2 Syntax*

**Descriptive**

!*prio_msg*

**Diagram**




### *7.1.3 Parameter*

**prio_msg** The priority message that will be routed to the system control program running under the Hercules Emulator.

### *7.1.4 Examples*

For a similar sample please see the “.reply” command.

## 7.2 # or * (Log comment to syslog)

### 7.2.1 Function

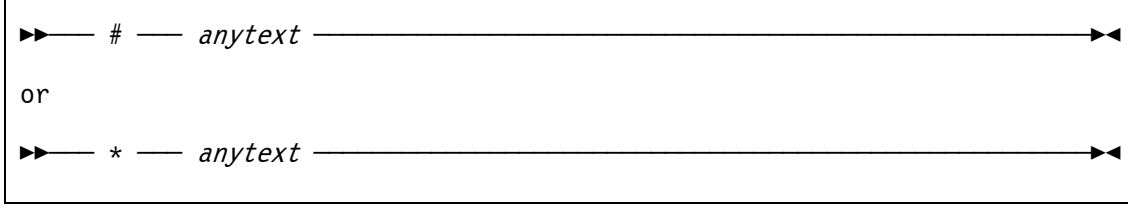
The hash ("#") or asterisk ("*") simply provide a convenient way of entering comments into the Hercules console log with no other effect. The comment is not processed in any way other than to echo it at the console.

### 7.2.2 Syntax

**Descriptive**

```
# anytext
   or
* anytext
```

**Diagram**




### 7.2.3 Parameter

**anytext** The text that has to be displayed on the Hercules system log.

### 7.2.4 Examples

**Example 1:**

Log a text string to the syslog.

```
09:09:34.781 00000FC4 * Comment routed to Hercules system log.
```

**Figure 9: Log comment command**

## 7.3 .reply (SCP command)

### 7.3.1 Function

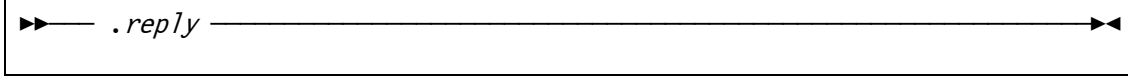
The ".reply" (SCP) command is used to give a reply to a system control program (i.e. the guest operating system running under Hercules) message. To reply to a SCP message that gets issued to the Hercules console, prefix the reply with a period (dot).

### 7.3.2 Syntax

**Descriptive**

*.reply*

**Diagram**




### 7.3.3 Parameter

**reply** The reply that will be routed to the system control program running under the Hercules Emulator.

### 7.3.4 Examples

**Example 1:**

Reply to message IGGN504A.

```
2007/06/30 09:09:34.781 00000FC4 *   IGGN504A SPECIFY UNIT FOR CATALOG.MVS.MASTER ON MVSRES
2007/06/30 09:10:04.390 00000FC4 .r 00,0148
2007/06/30 09:10:04.406 00000FC4 HHCCP041I SYSCONS interface active
2007/06/30 09:10:04.406 00000FC4  IEE600I REPLY TO 00 IS;0148
```

**Figure 10: SCP command**

## 7.4 ? (List all commands / command specific help)

### 7.4.1 Function

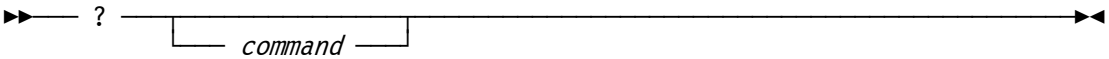
The "?" command is an alias for the HELP command and displays additional information for specific Hercules console commands if extra information is available. If the command has additional help text it will be displayed. "?" without parameter lists all available commands. The help text is usually limited to explaining the format of the command and its various required or optional parameters and is not meant to replace reading the documentation.

### 7.4.2 Syntax

**Descriptive**

? [*command*]

**Diagram**




### 7.4.3 Parameter

**command** The Hercules console command to which additional information is desired.

### 7.4.4 Examples

**Example 1:**

List all valid console commands.

```
22:51:42.150 00000F28 ?
22:51:42.150 00000F28 HHCPN140I Valid panel commands are...
22:51:42.150 00000F28
22:51:42.150 00000F28   Command      Description...
22:51:42.150 00000F28   -------      ------------------------------------------------
22:51:42.165 00000F28   ?            list all commands
22:51:42.165 00000F28   help         command specific help
22:51:42.165 00000F28
22:51:42.165 00000F28   *            (log comment to syslog)
22:51:42.165 00000F28
22:51:42.165 00000F28   hst          history of commands
22:51:42.165 00000F28   log          direct log output
22:51:42.165 00000F28   logopt       change log options
```

```
22:51:42.165 00000F28   version      display version information
22:51:42.165 00000F28
22:51:42.165 00000F28   quit         terminate the emulator
22:51:42.165 00000F28   exit         (synonym for 'quit')
22:51:42.165 00000F28
22:51:42.165 00000F28   cpu          define target cpu for panel display and commands
22:51:42.165 00000F28
22:51:42.165 00000F28   start        start CPU (or printer device if argument given)
22:51:42.165 00000F28   stop         stop CPU (or printer device if argument given)
22:51:42.165 00000F28
22:51:42.165 00000F28   startall     start all CPU's
22:51:42.165 00000F28   stopall      stop all CPU's
22:51:42.181 00000F28
22:51:42.181 00000F28   herclogo     Read a new hercules logo file
22:51:42.181 00000F28
22:51:42.181 00000F28   traceopt     Instruction trace display options
22:51:42.181 00000F28
22:51:42.181 00000F28   sf+dev       add shadow file
22:51:42.181 00000F28   sf-dev       delete shadow file
22:51:42.181 00000F28   sf=dev ..    rename shadow file
22:51:42.181 00000F28   sfc          compress shadow files
22:51:42.181 00000F28   sfd          display shadow file stats
.
.
.
22:51:42.181 00000F28
22:51:42.181 00000F28   t{+/-}dev    turn CCW tracing on/off
22:51:42.181 00000F28   s{+/-}dev    turn CCW stepping on/off
22:51:42.181 00000F28   t{+/-}CKD    turn CKD_KEY tracing on/off
22:51:42.181 00000F28   f{+/-}adr    mark frames unusable/usable
```

**Figure 11: “?” command**

## 7.5 AEA (Display AEA absolute-effective-address tables)

### *7.5.1 Function*

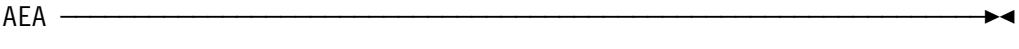
Display the Hercules AEA tables. The AEA table is an address lookup accelerator which saves absolute data addresses for further lookups to eliminate the calls to logical_to_main.This shortens the translation path length.

### *7.5.2 Syntax*

**Descriptive**

AEA

**Diagram**




### *7.5.3 Parameter*

None.

### *7.5.4 Examples*

**Example 1:**

Display the AEA tables

```
14:21:38.731 00000FC4 aea
14:21:38.731 00000FC4 aea mode   Primary
14:21:38.731 00000FC4 aea ar     0d 07 01 -1 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
14:21:38.731 00000FC4 aea common              0  0 01  0  0  0  0  0 01  0  0  0  0  0 01  0  0
14:21:38.731 00000FC4 aea cr[1]  000000003fffe07f
14:21:38.731 00000FC4     cr[7]  000000003fffe07f
14:21:38.731 00000FC4     cr[13] 000000003fffe07f
14:21:38.731 00000FC4     cr[r]  00000000ffffffff
```

**Figure 12: AEA command**

## 7.6 AIA (List AIA absolute-instruction-address fields)

### *7.6.1 Function*

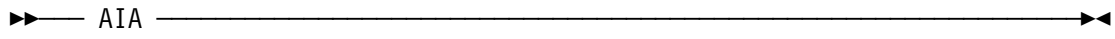
Display the Hercules AIA fields. The AIA fields are an instruction fetch accelerator which save absolute instruction addresses for further lookups to shorten the translation path length.

### *7.6.2 Syntax*

**Descriptive**

```
AIA
```

**Diagram**




### *7.6.3 Parameter*

None.

### *7.6.4 Examples*

**Example 1:**

Display the AIA fields.

```
00000FC4 aia
00000FC4 mainstor 10041000  aim 4ECA4000  aiv 0000000001581000  aie 0000000000000000
```

**Figure 13: AIA command**

## 7.7 AR (Display access registers)

### *7.7.1 Function*

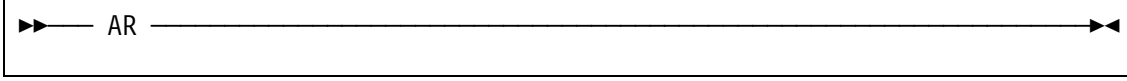
Display the current contents of the access registers.

### *7.7.2 Syntax*

**Descriptive**

AR

**Diagram**




### *7.7.3 Parameter*

None.

### *7.7.4 Examples*

**Example 1:**

Display the access registers.

```
14:22:01.091 00000FC4 ar
14:22:01.091 00000FC4 CPU0000: AR00=00000000  AR01=00000000  AR02=0000195B  AR03=88586380
14:22:01.091 00000FC4 CPU0000: AR04=88582860  AR05=090C52C8  AR06=88586190  AR07=00000000
14:22:01.091 00000FC4 CPU0000: AR08=8858631C  AR09=081F6060  AR10=00000000  AR11=00000000
14:22:01.091 00000FC4 CPU0000: AR12=00000000  AR13=00000000  AR14=00000002  AR15=00000000
```

**Figure 14: AR command**

## 7.8 ARCHMODE (Set architecture mode)

### *7.8.1 Function*

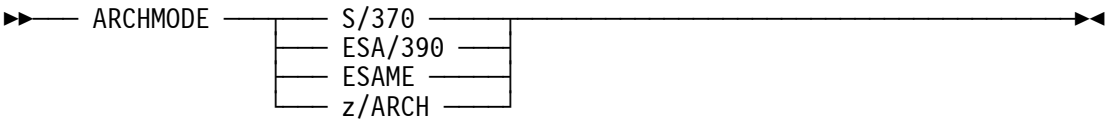
Specifying the archmode command without any parameters displays the current setting of the the initial architecture mode. Additional parameters may be specified to set a new value for the architecture mode.

### *7.8.2 Syntax*

**Descriptive**

```
ARCHMODE [S/370 | ESA/390 | ESAME | z/ARCH]
```

**Diagram**




### *7.8.3 Parameter*

**S/370** Use S/370 for OS/360, VM/370 and MVS 3.8

**ESA/390** Use ESA/390 for MVS/XA, MVS/ESA, OS/390, VM/ESA, VSE/ESA, Linux/390 and ZZSA. z/OS can be run until version 1.2 with ESA/390 mode without installed bimodal feature or until version 1.4 with installed bimodal feature.

**ESAME** Use ESAME (Enterprise System Architecture, Modal Extensions) for z/OS, z/VM, z/VSE and z/Linux. When ESAME is specified, the machine will always be IPL'ed in ESA/390 mode but the system is capable of being switched into the z/Architecture mode after IPL. This is handled automatically by all z/Architecture operating systems.

**z/ARCH** This is the same as ESAME.

### *7.8.4 Examples*

**Example 1:**

Display the current setting of the initial architecture mode.

```
2007/06/30 08:15:41.859 00000FC4 archmode
2007/06/30 08:15:41.859 00000FC4 HHCPN126I Architecture mode = ESA/390
```

**Figure 15: ARCHMODE command (display architecture mode)**

**Example 2:**

Change the setting of the current architecture mode.

```
2007/06/30 08:15:54.625 00000FC4 archmode esame
2007/06/30 08:15:54.625 00000FC4 HHCPN129I Architecture successfully set to z/Arch mode.
```

**Figure 16: ARCHMODE command (set architecture mode)**

## 7.9 ATTACH (Configure device)

### *7.9.1 Function*

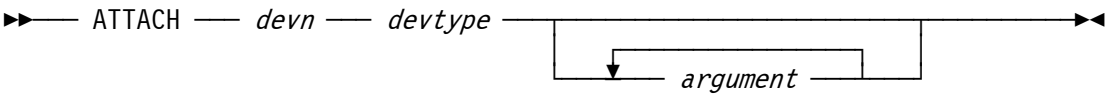
The ATTACH command configures a new device and makes it available in the active configuration. The effect of this command is immediately visible in the device display of the Hercules Windows GUI or the Hercules “Peripherals” display if the GUI is not used.

### *7.9.2 Syntax*

**Descriptive**

ATTACH *devn type* [*argument* [*argument* ...]]

**Diagram**




### *7.9.3 Parameter*

**devn** The device number of the device that is to be configured in the current configuration.

**type** The device type that is to be configured in the current configuration. For a list of valid device types see chapter 5 (Device Definition Descriptions).

**arguments** These are the additional arguments for the device that is to be configured in the current configuration. The arguments are dependent on the device type. For additional information on the valid arguments see chapter 5 (Device Definition Descriptions).

### *7.9.4 Examples*

**Example 1:**

Attach a printer device to the active configuration.

```
2007/06/30 13:32:45.562 00000618 attach 000f 1403 d:/mvs/prt/prt2.txt crlf
```

**Figure 17: ATTACH command (configure printer)**

**Example 2:**

Attach a display terminal to the active configuration.

```
2007/06/30 13:33:01.062 00000618 attach 0700 3270
```

**Figure 18: ATTACH command (configure display terminal)**

## 7.10 AUTOMOUNT (Show or update allowable tape automount directories)

### *7.10.1 Function*

The AUTOMOUNT command allows it to add or delete entries from the list of allowable/unallowable tape automount directories, or lists all currently defined list entries, if any.

The format of the <dir> directory operand for add/del operations is identical to that as described in the documentation for the AUTOMOUNT configuration file statement (i.e. prefix with '+' or '-' as needed).

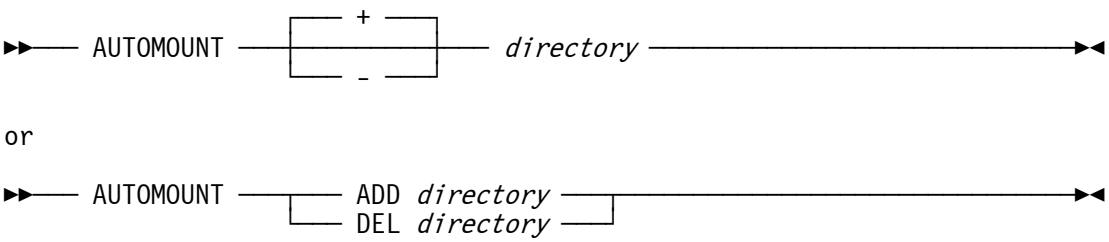
The automount feature is approriately enabled or disabled for all tape devices as needed depending on the updated empty/non-empty list state.

### *7.10.2 Syntax*

**Descriptive**

```
AUTOMOUNT {ADD directory | DEL directory | LIST}
   or
AUTOMOUNT [ + | - ]directory
```

**Diagram**




### *7.10.3 Parameter*

**ADD or +** Add an entry to the list of allowable tape automount directories.

**DEL or -** Delete an entry from the list of allowable tape automount directories.

**LIST** List all currently defined list entries of allowable tape automount directories.

**directory** Specifies the host system directory where the guest is allowed or not to automatically load virtual tape volumes from.

### *7.10.4 Examples*

**Example 1:**

Add an entry to the list of allowable tape automount directories.

```
0000097C automount add D:/MVS/Tape_1
0000097C HHCPN210I Default Allowed AUTOMOUNT directory = "D:\MVS\Tape_1\"
0000097C automount add D:/MVS/Tape_2
0000097C HHCPN210I Allowed AUTOMOUNT directory = "D:\MVS\Tape_2\"
```

**Figure 19: AUTOMOUNT command (add entry)**

**Example 2:**

List all currently defined list entries of allowable tape automount directories.

```
0000097C automount list
0000097C HHCPN203I "+D:\MVS\Tape_1\"
0000097C HHCPN203I "+D:\MVS\Tape_2\"
```

**Figure 20: AUTOMOUNT command (list entries)**

**Example 3:**

Delete an entry from the list of allowable tape automount directories.

```
0000097C automount del D:/MVS/Tape_2
0000097C HHCPN214I Ok.
```

**Figure 21: AUTOMOUNT command (delete entry)**

## 7.11 AUTO_SCSI_MOUNT (Automatic SCSI tape mounts)

### *7.11.1 Function*

The AUTO_SCSI_MOUNT command displays or modifies the automatic SCSI tape mount option. When entered without any operands it displays the current value and any pending tape mount requests. Entering a value between 1 and 99 enables the option and specifies how often (in seconds) to query SCSI tape drives to automatically detect when a tape has been mounted. When a tape is mounted an unsolicited device attention interrupt will be presented to the guest operating system.

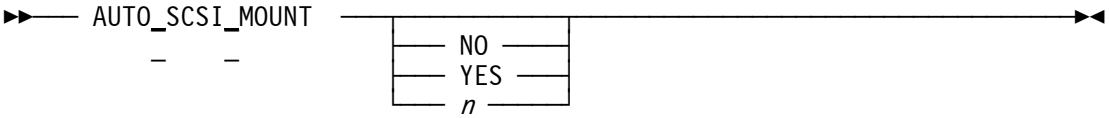
Notes: Enabling this option may negatively impact Hercules performance depending on how the host operating system (Windows, Linux, etc.) processes SCSI attached tape drive status queries. AUTO_SCSI_MOUNT is a synonym for the SCSIMOUNT command.

### *7.11.2 Syntax*

**Descriptive**

```
AUTO_SCSI_MOUNT [NO | YES | n]
```

**Diagram**




### *7.11.3 Parameter*

**NO** No indicates that the AUTO_SCSI_MOUNT option is disabled forcing all SCSI tape mounts to be done manually via an appropriate DEVINIT command.

**YES** Yes enables the option and causes periodic queries of the SCSI tape drive in a five second interval to automatically detect when a tape is mounted. YES is equivalent with “AUTO_SCSI_MOUNT 5”.

**n** A value from 1 to 99 seconds inclusive enables the option and causes periodic queries of the SCSI tape drive to automatically detect when a tape is mounted.

### *7.11.4 Examples*

**Example 1:**

Display the current AUTO_SCSI_MOUNT settings.

```
2009/03/07 21:50:43.843 00000574 auto_scsi_mount
2009/03/07 21:50:43.843 00000574 Auto SCSI mount 0 seconds
```

**Figure 22: AUTO_SCSI_MOUNT command (display settings)**

**Example 2:**

Set AUTO_SCSI_MOUNT to 30 seconds.

```
2009/03/07 21:50:52.062 00000574 auto_scsi_mount 30
2009/03/07 21:51:04.468 00000574 auto_scsi_mount
2009/03/07 21:51:04.468 00000574 Auto SCSI mount 30 seconds
```

**Figure 23: AUTO_SCSI_MOUNT command (set value)**

## 7.12 B (Set breakpoint)

### *7.12.1 Function*

The B command sets a breakpoint. The argument gives an instruction address or a range of addresses where you wish to halt the execution. Once the breakpoint is reached, instruction execution is temporarily halted and the next instruction to be executed is displayed. You may then examine registers and / or storage etc. To continue execution after reaching a breakpoint enter the “G” command.

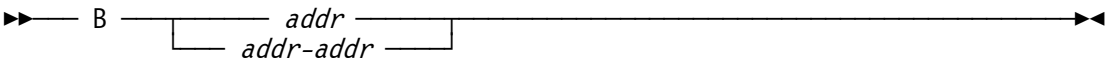
The “B” and “B+” commands are synonymous with the “S+” command.

### *7.12.2 Syntax*

**Descriptive**

B {*addr* | *addr-addr*}

**Diagram**




### *7.12.3 Parameter*

**addr** Instruction address or range of addresses where instruction execution is to be stopped.

### *7.12.4 Examples*

**Example 1:**

Set a breakpoint at instruction address 01000000.

```
12:50:14.546 00000FC4 b 01000000
12:50:14.546 00000FC4 HHCPN040I Setting breakpoint at 0000000001000000-0000000001000000
```

**Figure 24: B command**

## 7.13 B+ (Set breakpoint)

### *7.13.1 Function*

The B+ command sets a breakpoint. The argument gives an instruction address or a range of addresses where you wish to halt the execution. Once the breakpoint is reached, instruction execution is temporarily halted and the next instruction to be executed is displayed. You may then examine registers and / or storage etc. To continue execution after reaching a breakpoint enter the “G” command.

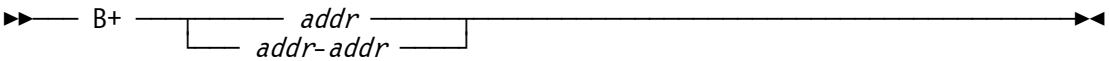
The “B” and “B+” commands are synonymous with the “S+” command.

### *7.13.2 Syntax*

**Descriptive**

B+ {*addr* | *addr-addr*}

**Diagram**




### *7.13.3 Parameter*

**addr** Instruction address or range of addresses where instruction execution is to be stopped.

### *7.13.4 Examples*

**Example 1:**

Set a breakpoint at instruction address 01000000.

```
12:50:14.546 00000FC4 b+ 01000000
12:50:14.546 00000FC4 HHCPN040I Setting breakpoint at 0000000001000000-0000000001000000
```

**Figure 25: B+ command**

## 7.14 B- (Delete breakpoint)

### *7.14.1 Function*

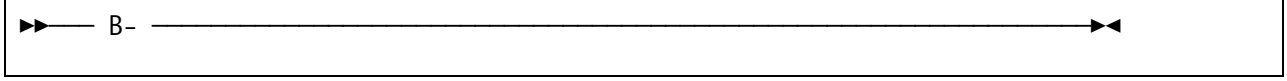
The B- command removes any previously set breakpoint.

### *7.14.2 Syntax*

**Descriptive**

B-

**Diagram**




### *7.14.3 Parameter*

None.

### *7.14.4 Examples*

**Example 1:**

Delete the previously set breakpoint.

```
2007/06/30 12:50:33.312 00000FC4 b-
2007/06/30 12:50:33.312 00000FC4 HHCPN041I Deleting breakpoint
```

**Figure 26: B- command**

## 7.15 CACHE (Cache command)

### *7.15.1 Function*

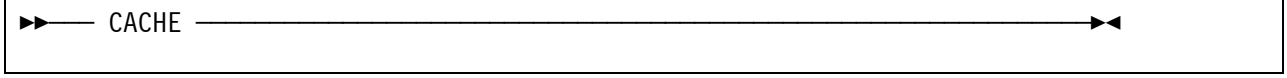
The CACHE command displays the current Hercules cache statistics.

### *7.15.2 Syntax*

**Descriptive**

```
CACHE
```

**Diagram**




### *7.15.3 Parameter*

None.

### *7.15.4 Examples*

**Example 1:**

Display the current cache statistics.

```
2007/06/30 14:28:48.622 00000FC4 cache
2007/06/30 14:28:48.622 00000FC4 cache............          0
2007/06/30 14:28:48.622 00000FC4 nbr .............        229
2007/06/30 14:28:48.622 00000FC4 busy ............          0
2007/06/30 14:28:48.622 00000FC4 busy% ...........          0
2007/06/30 14:28:48.622 00000FC4 empty ...........          0
2007/06/30 14:28:48.622 00000FC4 waiters .........          0
2007/06/30 14:28:48.622 00000FC4 waits ...........          0
2007/06/30 14:28:48.622 00000FC4 buf size ........   13014528
2007/06/30 14:28:48.622 00000FC4 hits ............      80529
2007/06/30 14:28:48.622 00000FC4 fast hits .......      73072
2007/06/30 14:28:48.622 00000FC4 misses ..........      22839
2007/06/30 14:28:48.622 00000FC4 hit% ............         77
2007/06/30 14:28:48.622 00000FC4 age .............      94500
2007/06/30 14:28:48.622 00000FC4 last adjusted ... Thu Jan 01 01:00:00 1970
2007/06/30 14:28:48.622 00000FC4 last wait ....... Thu Jan 01 01:00:00 1970
2007/06/30 14:28:48.622 00000FC4 adjustments .....          0
2007/06/30 14:28:48.622 00000FC4 cache[1] ....... not created
```

```
2007/06/30 14:28:48.622 00000FC4 cache[2] ....... not created
2007/06/30 14:28:48.622 00000FC4 cache[3] ....... not created
2007/06/30 14:28:48.622 00000FC4 cache[4] ....... not created
2007/06/30 14:28:48.622 00000FC4 cache[5] ....... not created
2007/06/30 14:28:48.622 00000FC4 cache[6] ....... not created
2007/06/30 14:28:48.622 00000FC4 cache[7] ....... not created
```

**Figure 27: CACHE command**

## 7.16 CCKD (CCKD command)

### *7.16.1 Function*

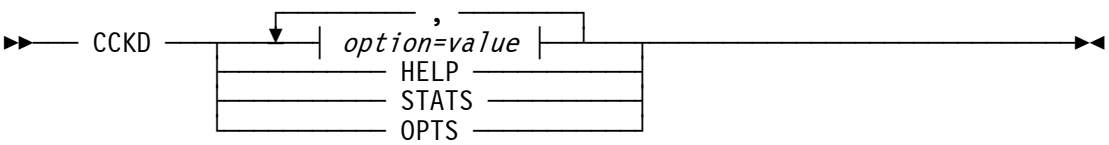
The CCKD console command can be used to affect CCKD processing. The CCKD console command supports the same options as the CCKD configuration statement.

### *7.16.2 Syntax*

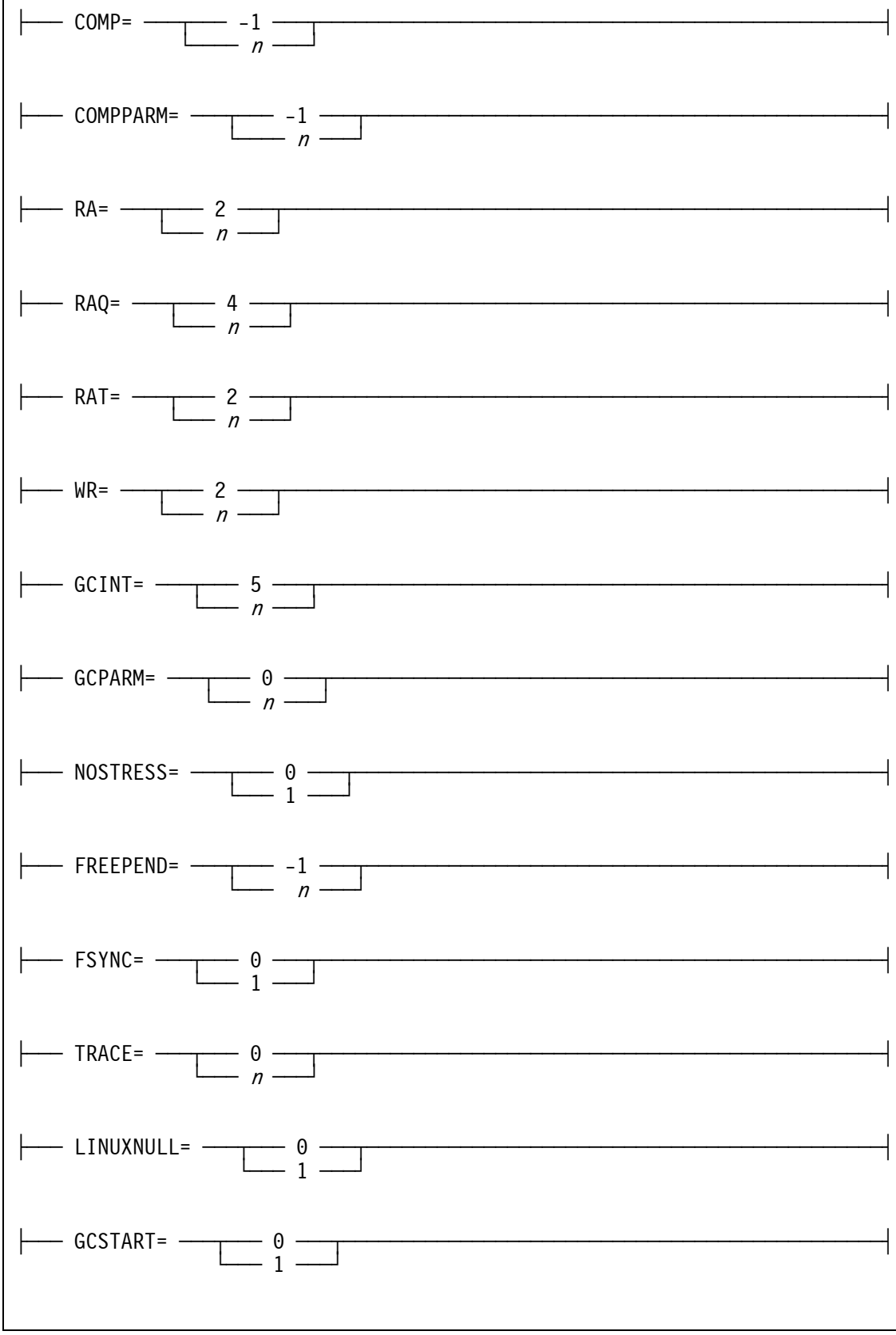
**Descriptive**

```
CCKD [HELP | STATS | OPTS | option=value [,option=value ... ]]

   where option can be:

[COMP={-1 | n}]
[,COMPPARM={-1 | n}]
[,RA={2 | n}]
[,RAQ={4 | n}]
[,RAT={2 | n}]
[,WR={2 | n}]
[,GCINT={5 | n}]
[,GCPARM={0 | n}]
[,NOSTRESS={0 | 1}]
[,FREEPEND={-1 | n}]
[,FSYNC={0 | 1}]
[,TRACE={0 | n}]
[,LINUXNULL={0 | 1}]
[,GCSTART={0 | 1}]
```

**Diagram**



where option can be:



### *7.16.3 Parameter*

**HELP** Display the CCKD help information (CCKD command only).

**STATS** Display the current CCKD statistics (CCKD command only).

**OPTS** Display the current CCKD options (CCKD command only).

**option** Set a CCKD option. Multiple options may be specified, separated by commas with no intervening blanks.

**The CCKD options are:**

**COMP=n** Specifies the compression type to be used. This overrides the compression used for all CCKD files. The default (-1) means don't override the compression. Valid compression types are:

**-1** Default

**0** None

**1** zlib

**2** bzip2

**COMPPARM=n** Overrides the compression parameter. A higher value generally means more compression at the expense of CPU and/or storage. The default (-1) means don't override the compression parameter. The value of *n* can be from -1 and 9.

**RA=n** Sets the Number of read ahead threads. When sequential track or block group access is detected, some number (RAT=n) of tracks or block groups are queued (RAQ=n) to be read by one of the read ahead threads. The default is 2, the value of *n* can be a number from 1 to 9.

**RAQ=n** Sets the size of the read ahead queue. When sequential track or block group access is detected, some number (RAT=n) of tracks or block groups are queued in the read ahead queue. The default is 4, the value of *n* can be a number from 0 to 16. A value of zero disables read ahead.

**RAT=n** Sets the number of tracks or block groups to read ahead when sequential track or block group access is detected. The default is 2, the value of ratn can be a number from 0 to 16. A value of zero disables read ahead.

**WR=n** Sets the number of writer threads. When the cache is flushed, updated cache entries are marked write pending and a writer thread is signalled. The writer thread compresses the track or block group and writes the compressed image to the emulation file.

A writer thread is CPU-intensive while compressing the track or block group and I/O-intensive while writing the compressed image. The writer thread runs one nicer than the CPU thread(s). The default is 2, a value from 1 to 9 can be specified

**GCINT=n** This is the number of seconds the garbage collector thread waits during an interval. At the end of an interval the garbage collector performs space recovery, flushes the cache and optionally 'fsyncs' the emulation file. However, the file will not be 'fsynced' unless at least 5 seconds have elapsed since the last synchronization (FSYNC). The default is 10 seconds. You can specify a number between 1 and 60.

**GCPARM=n** A value affecting the amount of data moved during the garbage collectors space recovery routine. The garbage collector determines an amount of space to move based on the ratio of free space to used space in an emulation file and on the number of free spaces in the file. The garbage collector wants to reduce the free space to used space ratio and the number of free spaces.

The value is logarithmic; a value of 8 means moving $2^8$ the selected value while a negative value similarly decreases the amount to be moved. Normally, 256K will be moved for a file in an interval. Specifying a value of 8 can increase the amount to 64M. At least 64K will be moved. Specifying a large value (such as 8) may not increase the garbage collection efficiency correspondingly. The default is 0. You can specify a number from -8 to 8.

**NOSTRESS=n** Indicates whether stress writes will occur or not. A track or block group may be written under stress when a high percentage of the cache is pending write or when a device I/O thread is waiting for a cache entry. When a stressed write occurs, the compression algorithm and/or compression parm may be relaxed, resulting in faster compression but usually a larger compressed image. If NOSTRESS is set to one, then a stressed situation is ignored. You would typically set this value to one when you want create the smallest emulation file possible in exchange for a possible performance degradation. The default is 0. You can specify 0 (enable stressed writes) or 1 (disable stressed writes).

**FREEPEND=n** Specifies the free pending value for freed space. When a track or block group image is written, the space it previously occupied is freed. This space will not be available for future allocations until *n* garbage collection intervals have completed. In the event of a catastrophic failure, previously written track or block group images should be recoverable if the current image has not yet been written to the physical disk. By default the value is set to -1 which means that if FSYNC is specified then the value is 1 otherwise it is 2. If 0 is specified then freed space is immediately available for new allocations. The default is -1. You can specify a number from -1 to 4.

**FSYNC=n** Enables or disables FSYNC. When FSYNC is enabled then the disk emulation file is synchronized with the physical hard disk at the end of a garbage collection interval (no more often than 5 seconds though). This means that if FREEPEND is non-zero and a catastrophic error occurs, the emulated disks should be recovered coherently. However, FSYNC may cause performance degradation depending on the host operating system and / or the host operating system level. The default is 0 (fsync disabled), you can specify 0 (disable FSYNC) or 1 (enable FSYNC).

**TRACE=n** value Specifies the number of CCKD trace entries. You would normally specify a non-zero when debugging or capturing a problem in CCKD code. When the problem occurs, you should enter the "k" Hercules console command which will print the trace table entries. The default is 0. You can specify a number between 0 and 200000. Each trace entry represents 128 bytes. Normally, for debugging, it is recommended to use 100000.

**LINUXNULL=n** If set to 1 then tracks written to 3390 CCKD volumes that were initialized with the *-linux* option will be checked if they are null (that is if all 12 4096 byte user records contain zeroes). This is used by the DASDCOPY utility. The default is 0.

**GCSTART=n** If set to 1 then space recovery will become active on any emulated disks that have free space. Normally space recovery will ignore emulated disks until they have been updated. The default is 0.

**Notes:**

- *raq* should be at least as large as *ra*. Read ahead threads are scheduled from entries in the read ahead queue. Likewise *rat* should not exceed *raq* because only *raq* tracks or block groups can be queued at any time.

- The number of writer threads *wr* should usually be 1 more than the number of host processors. This is because one writer thread could be CPU-bound (compressing a track or block-group image) and the other could be i/o-bound (writing the compressed image).

- The garbage collection interval governs the maximum time in seconds an updated track or block group image will reside in storage before being written to the emulation file. A large value may mean more data loss if a catastrophic error occurs. A small value may mean that more CPU time is spent compressing images.

For example, suppose that a particular image is updated several times each second. If the interval is changed from the default 5 seconds to 1 second, then that image will be compressed and written 5 times more frequently. A large value may cause more cache flushes within a garbage collection interval. These kinds of flushes mean that a read will wait because there are no available cache entries, slowing the emulated operating system. A large value will also cause more pending free space to build up (since free space is flushed each interval). This may mean that the garbage collector space recovery routine will perform more work and the resulting emulation file may be larger.

- Specify *fsync=1* and *gcint=5* if you are seriously concerned about your data being lost due to a failure. *fsync* will ensure your data on disk is coherent. However, fsync may cause a noticeable performance degradation. Note that an fsync will not be performed more often than every 5 seconds.

## *7.16.4 Examples*

**Example 1:**

Display the CCKD help panel. There is no specific CCKD help command. The CCKD command without argument will present the list of available options.

```
17:51:36.796 0000008E cckd
17:51:36.796 0000008E cckd command parameters:
17:51:36.796 0000008E help            Display help message
17:51:36.796 0000008E stats           Display cckd statistics
17:51:36.796 0000008E opts            Display cckd options
17:51:36.796 0000008E comp=<n>                Override compression           (-1 .. 2)
17:51:36.796 0000008E compparm=<n>    Override compression parm              (-1 .. 9)
17:51:36.796 0000008E ra=<n>          Set number readahead threads           (1 .. 9)
17:51:36.796 0000008E raq=<n>         Set readahead queue size               (0 .. 16)
17:51:36.796 0000008E rat=<n>         Set number tracks to read ahead        (0 .. 16)
17:51:36.796 0000008E wr=<n>          Set number writer threads              (1 .. 9)
17:51:36.796 0000008E gcint=<n>       Set garbage collector interval (sec)   (1 .. 60)
17:51:36.796 0000008E gcparm=<n>      Set garbage collector parameter        (-8 .. 8)
17:51:36.796 0000008E                     (least agressive ... most aggressive)
17:51:36.796 0000008E nostress=<n>    1=Disable stress writes
17:51:36.796 0000008E freepend=<n>    Set free pending cycles                (-1 .. 4)
17:51:36.796 0000008E fsync=<n>       1=Enable fsync()
17:51:36.796 0000008E trace=<n>       Set trace table size                   (0 .. 200000)
```

**Figure 28: CCKD HELP command**

**Example 2:**

Display the CCKD options panel.

```
17:51:13.453 0000008E cckd opts
17:51:13.453 0000008E comp=-1,compparm=-1,ra=2,raq=4,rat=2,wr=2,gcint=10,gcparm=0,nostress=0,
17:51:13.453 0000008E         freepend=-1,fsync=0,trace=0,linuxnull=0
```

**Figure 29: CCKD OPTS command**

**Example 3:**

Display the CCKD statistics panel.

```
0000008E cckd stats
0000008E reads....       344 Kbytes...      5175 writes...         0 Kbytes...         0
0000008E readaheads       42 misses...         4 syncios..       337 misses...       302
0000008E switches.       650 l2 reads.         6              stress writes...         0
0000008E cachehits       347 misses...       303 l2 hits..        33 misses...         6
0000008E waits                                   i/o......         2 cache....         0
0000008E garbage collector   moves....         0 Kbytes...         0
```

**Figure 30: CCKD STATS command**

**Example 4:**

Set a CCKD parameter.

```
18:23:38.187 00000B28 cckd gcint=30
18:23:38.187 00000B28 comp=-1,compparm=-1,ra=2,raq=4,rat=2,wr=2,gcint=30,gcparm=0,nostress=0,
18:23:38.187 00000B28         freepend=-1,fsync=0,trace=0,linuxnull=0
```

**Figure 31: CCKD command (set options)**

## 7.17 CD (Change directory)

### *7.17.1 Function*

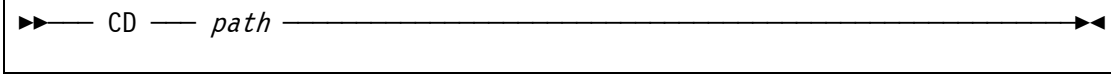
The CD command changes the current directory.

### *7.17.2 Syntax*

**Descriptive**

CD *path*

**Diagram**




### *7.17.3 Parameter*

**path** This specifies the new path. The path can be specified absolute (“D:\S390\DASD”) or relative to the current path (“DASD”).

### *7.17.4 Examples*

**Example 1:**

Change the current directory (absolute path).

```
14:22:01.091 00000FC4 cd D:\S390
14:22:01.091 00000FC4 D:\S390
```

**Figure 32: CD command (absolute path)**

**Example 2:**

Change the current directory (relative path).

```
14:22:55.084 00000FC4 cd DASD
14:22:55.084 00000FC4 D:\S390\DASD
```

**Figure 33: CD command (relative path)**

## 7.18 CF (Configure CPU online or offline)

### *7.18.1 Function*

The CF command is used to place a CPU online or offline. The number of the CPU to be taken online or offline has to be specified first with the CPU command (refer to the CPU command for further information). If the CF command is issued without a parameter, the actual status of the CPU is displayed.

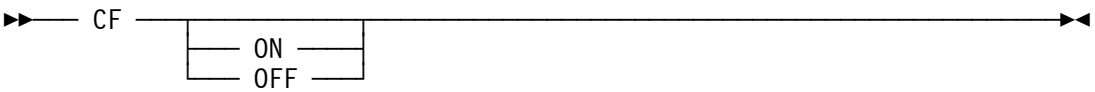
Use the CFALL command to display the status of all CPUs or to configure all CPUs online or offline.

### *7.18.2 Syntax*

**Descriptive**

```
CF [ON | OFF]
```

**Diagram**




### *7.18.3 Parameter*

**ON** Place the specified CPU online.

**OFF** Place the specified CPU offline.

### *7.18.4 Examples*

**Example 1:**

Display the actual status of CPU 1.

```
2007/06/30 14:12:38.531 00000FC4 cpu 1
2007/06/30 14:12:42.796 00000FC4 cf
2007/06/30 14:12:42.796 00000FC4 HHCPN152I CPU0001 online
```

**Figure 34: CF command (display CPU status)**

**Example 2:**

Configure CPU 1 offline.

```
2007/06/30 14:12:47.531 00000FC4 cpu 1
2007/06/30 14:12:49.859 00000FC4 cf off
2007/06/30 14:12:49.859 00000FC4 HHCCP008I CPU0001 thread ended: tid=00000D28, pid=2972
2007/06/30 14:12:49.859 00000FC4 HHCPN153I CPU0001 offline
```

**Figure 35: CF command (configure CPU offline)**

**Example 3:**

Configure CPU 1 online.

```
14:12:50.531 00000FC4 cpu 1
14:12:52.906 00000FC4 cf on
14:12:52.906 00000FC4 HHCCP002I CPU0001 thread started: tid=000009D8, pid=2972, priority=0
14:12:52.906 00000FC4 HHCCP003I CPU0001 architecture mode ESA/390
14:12:52.906 00000FC4 HHCPN152I CPU0001 online
```

**Figure 36: CF command (configure CPU online)**

## 7.19 CFALL (Configure all CPUs online or offline)

### *7.19.1 Function*

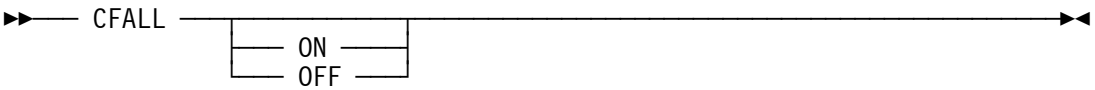
The CFALL command is used to place all CPUs online or offline. If the CFALL command is issued without a parameter, the actual status of all CPUs is displayed

### *7.19.2 Syntax*

**Descriptive**

```
CFALL [ON | OFF]
```

**Diagram**




### *7.19.3 Parameter*

**ON** Place all CPUs online.

**OFF** Place all CPUs offline.

### *7.19.4 Examples*

**Example 1:**

Display the current status of all CPUs.

```
2007/06/30 14:24:25.875 00000FC4 cfall
2007/06/30 14:24:25.875 00000FC4 HHCPN154I CPU0000 online
2007/06/30 14:24:25.875 00000FC4 HHCPN154I CPU0001 online
2007/06/30 14:24:25.875 00000FC4 HHCPN155I CPU0002 offline
2007/06/30 14:24:25.875 00000FC4 HHCPN155I CPU0003 offline
2007/06/30 14:24:25.875 00000FC4 HHCPN155I CPU0004 offline
2007/06/30 14:24:25.875 00000FC4 HHCPN155I CPU0005 offline
```

**Figure 37: CFALL command (display status of all CPUs)**

**Example 2:**

Configure all CPUs offline.

```
2007/06/30 14:24:46.625 00000FC4 cfall off
2007/06/30 14:24:46.625 00000FC4 HHCCP008I CPU0000 thread ended: tid=00000CC0, pid=3172
2007/06/30 14:24:46.625 00000FC4 HHCCP008I CPU0001 thread ended: tid=0000012C, pid=3172
2007/06/30 14:24:46.625 00000FC4 HHCCP008I CPU0002 thread ended: tid=00000390, pid=3172
2007/06/30 14:24:46.625 00000FC4 HHCCP008I CPU0003 thread ended: tid=00000E08, pid=3172
2007/06/30 14:24:46.625 00000FC4 HHCCP008I CPU0004 thread ended: tid=00000EF4, pid=3172
2007/06/30 14:24:46.625 00000FC4 HHCCP008I CPU0005 thread ended: tid=00000FAC, pid=3172
2007/06/30 14:24:46.625 00000FC4 HHCPN155I CPU0000 offline
2007/06/30 14:24:46.625 00000FC4 HHCPN155I CPU0001 offline
2007/06/30 14:24:46.625 00000FC4 HHCPN155I CPU0002 offline
2007/06/30 14:24:46.625 00000FC4 HHCPN155I CPU0003 offline
2007/06/30 14:24:46.625 00000FC4 HHCPN155I CPU0004 offline
2007/06/30 14:24:46.625 00000FC4 HHCPN155I CPU0005 offline
2007/06/30 14:24:46.640 00000FC4 HHCTT003I Timer thread ended
```

**Figure 38: CFALL command (configure all CPUs offline)**

**Example 3:**

Configure all CPUs online.

```
14:24:42.375 00000FC4 cfall on
14:24:42.375 00000FC4 HHCCP002I CPU0000 thread started: tid=00000CC0, pid=3172, priority=0
14:24:42.375 00000FC4 HHCTT002I Timer thread started: tid=00000A14, pid=3172, priority=-20
14:24:42.375 00000FC4 HHCCP003I CPU0000 architecture mode ESA/390
14:24:42.375 00000FC4 HHCCP002I CPU0001 thread started: tid=0000012C, pid=3172, priority=0
14:24:42.375 00000FC4 HHCCP003I CPU0001 architecture mode ESA/390
14:24:42.375 00000FC4 HHCCP002I CPU0002 thread started: tid=00000390, pid=3172, priority=0
14:24:42.375 00000FC4 HHCCP003I CPU0002 architecture mode ESA/390
14:24:42.375 00000FC4 HHCCP002I CPU0003 thread started: tid=00000E08, pid=3172, priority=0
14:24:42.375 00000FC4 HHCCP003I CPU0003 architecture mode ESA/390
14:24:42.375 00000FC4 HHCCP002I CPU0004 thread started: tid=00000EF4, pid=3172, priority=0
14:24:42.375 00000FC4 HHCCP003I CPU0004 architecture mode ESA/390
14:24:42.375 00000FC4 HHCCP002I CPU0005 thread started: tid=00000FAC, pid=3172, priority=0
14:24:42.375 00000FC4 HHCCP003I CPU0005 architecture mode ESA/390
14:24:42.375 00000FC4 HHCPN154I CPU0000 online
14:24:42.375 00000FC4 HHCPN154I CPU0001 online
14:24:42.375 00000FC4 HHCPN154I CPU0002 online
14:24:42.375 00000FC4 HHCPN154I CPU0003 online
14:24:42.375 00000FC4 HHCPN154I CPU0004 online
14:24:42.375 00000FC4 HHCPN154I CPU0005 online
```

**Figure 39: CFALL command (configure all CPUs online)**

## 7.20 CLOCKS (Display TOD clock and CPU timer)

### *7.20.1 Function*

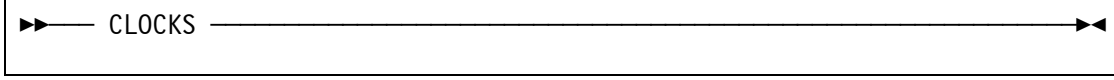
The CLOCKS command is used to display the actual values of the emulators various internal clocks and timers.

### *7.20.2 Syntax*

**Descriptive**

```
CLOCKS
```

**Diagram**




### *7.20.3 Parameter*

None.

### *7.20.4 Examples*

**Example 1:**

Display the TOD clock and CPU timer.

```
2007/06/30 14:22:22.137 00000FC4 clocks
2007/06/30 14:22:22.137 00000FC4 HHCPN028I tod = BE23401EA237A000    2005.364 13:22:22.115706
2007/06/30 14:22:22.137 00000FC4           h/w = BE23401EA237A000    2005.364 13:22:22.115706
2007/06/30 14:22:22.137 00000FC4           off = 0000000000000000
2007/06/30 14:22:22.137 00000FC4           ckc = BE23401EA77DD000    2005.364 13:22:22.137309
2007/06/30 14:22:22.137 00000FC4           cpt = 00000000000ED000
```

**Figure 40: CLOCKS command**

## 7.21 CMDTGT (Specify the command target)

### *7.21.1 Function*

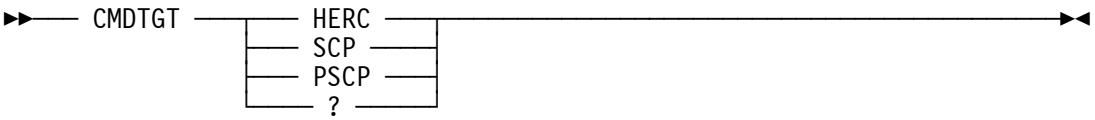
The CMDTGT command specifies the target environment for all following commands entered on the command line of the Hercules hardware console.

### *7.21.2 Syntax*

**Descriptive**

```
CMDTGT {HERC | SCP | PSCP | ?}
```

**Diagram**




### *7.21.3 Parameter*

**HERC** The command target is the Hercules Emulator. This is the default, when no CMDTGT target has been given.

**SCP** The target for the commands is the system control program (i.e. the guest operating system running under Hercules).

**PSCP** All entered commands are system control program (i.e. guest operating system) priority commands.

**?** Display the current target environment for commands.

### *7.21.4 Examples*

**Example 1:**

Display the current target environment for commands.

```
2008/08/30 09:10:04.390 00000FC4 cmdtgt ?
2008/08/30 09:10:04.390 00000FC4 cmdtgt: Commands are sent to Hercules
```

**Figure 41: CMDTGT command (display current setting)**

**Example 2:**

Set the system control program as the target for commands.

```
2008/08/30 09:10:04.390 00000FC4 cmdtgt scp
2008/08/30 09:10:04.390 00000FC4 cmdtgt: Commands are sent to scp
```

**Figure 42: CMDTGT command (set target to SCP)**

**Example 3:**

Set the command target from SCP or PSCP back to Hercules (see also HERC command).

```
2008/08/30 09:10:04.390 00000FC4 herc cmdtgt herc
2008/08/30 09:10:04.390 00000FC4 cmdtgt: Commands are sent to Hercules
```

**Figure 43: CMDTGT command (set target back to Hercules)**

## 7.22 CONKPALV (Display / alter console TCP/IP keep-alive settings)

### 7.22.1 Function

The CONKPALV command specifies the tn3270 console and telnet clients keepalive option values that control automatic detection of disconnected tn3270/telnet client sessions.

This is a built-in feature of TCP/IP and allows detection of unresponsive TCP/IP connections and not idle clients. That is to say, your connection will not be terminated after 3 seconds of idle time. Your 3270 session can remain idle for many minutes or hours or days without any data being transmitted. If the TCP/IP stack at the other end of the connection (not your 3270 client itself) fails to respond to the internal keepalive probe packets however, then it means that the TCP/IP stack itself is down or there has been a physical break in the connection.

Thus, even if your 3270 client is completely idle, your system's TCP/IP stack itself should still respond to the keepalive probes sent by the TCP/IP stack at the Hercules end of the link. If it doesn't, then TCP/IP will terminate the tn3270/telnet session which will cause Hercules to disconnect the terminal.

The three values can also be modified on-demand via the conkpalv panel command, which has the exact same syntax. Note that the syntax is very unforgiving: no spaces are allowed anywhere within the parentheses and each value must be separated from the other with a single comma.

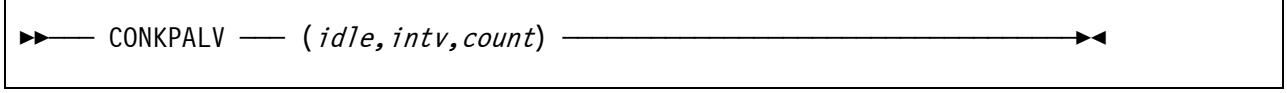
Please also note that not all systems support being able to modify all three values. That is, not all values may be able to be changed, and it is system dependent which values you can change and which values you cannot. On Windows for example, the *count* value is ignored and cannot be changed from its default value of 10. Other systems may ignore one or more or all three values and use platform defaults instead. This is entirely system dependent. Check you system's documentation for details regarding which values can be changed and which cannot as well as how to adjust your system's default values.

### 7.22.2 Syntax

**Descriptive**

```
CONKPALV (idle,intv,count)
```

**Diagram**




### 7.22.3 Parameter

**idle** The *idle* value specifies the timeout value in seconds with no activity until the first keepalive probe is sent. The default for the *idle* value is 3 seconds.

**intv** The *intv* value specifies the interval in seconds between the probes when no acknowledgement is received from the previous one. The default for the *intv* value is 1 second.

**count** The *count* value specifies the number of unacknowledged keep-alive packets sent before the connection is considered to have failed. The default value for the *count* value is 9 for non-Windows platforms and 10 for Windows systems.

*Note:* On Windows platforms the *count* value is ignored and cannot be changed from its default value of 10.

## *7.22.4 Examples*

**Example 1:**

Set the TCP/IP keep-alive settings to 5 seconds idle time, 3 seconds interval between the probes and disconnect after 15 consecutive failed probes.

```
2007/06/30 19:00:30.500 000000E8 conkpalv (5,3,15)
2007/06/30 19:00:30.500 000000E8 HHCPN190I Keep-alive = (5,3,15)
```

**Figure 44: CONKPALV command**

## 7.23 CPU (Define target CPU for console displays and commands)

### *7.23.1 Function*

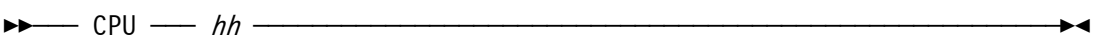
The CPU command sets the target CPU for console displays and commands in a Hercules multi-CPU environment. If the Hercules configuration file sets a single CPU environment, then no CPU command is necessary. If the NUMCPU system parameter in the Hercules configuration file is set greater than one and no CPU command is given, then always the first CPU (CPU 0) is the target for console displays and commands.

### *7.23.2 Syntax*

**Descriptive**

CPU *hh*

**Diagram**




### *7.23.3 Parameter*

**hh** The hexadecimal CPU address of the CPU in the multiprocessor configuration which you wish all console commands to apply to. For example, entering “CPU 1” followed by the command “GPR” in a dual CPU configuration will display the general purpose registers for CPU #1 in the configuration, as opposed to CPU #0.

### *7.23.4 Examples*

**Example 1:**

Display the general purpose registers for CPU #0 in a multiprocessor configuration.

```
16:54:24.781 00000FC4 cpu 0
16:54:31.625 00000FC4 gpr
16:54:31.625 00000FC4 CPU0000: GR00=00000000  GR01=00000000  GR02=0252E12C  GR03=02501EB0
16:54:31.625 00000FC4 CPU0000: GR04=02502400  GR05=00000C00  GR06=01379E76  GR07=00000000
16:54:31.625 00000FC4 CPU0000: GR08=0248EF58  GR09=00000000  GR10=0252EF7F  GR11=0252DF80
16:54:31.625 00000FC4 CPU0000: GR12=813799C8  GR13=0252DF80  GR14=81379D22  GR15=00000C00
```

**Figure 45: CPU command**

## 7.24 CR (Display or alter control registers)

### *7.24.1 Function*

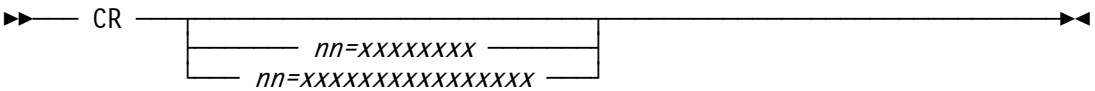
The CR command displays or alters the actual contents of the control registers.

### *7.24.2 Syntax*

**Descriptive**

CR [*nn=xxxxxxxx* | *nn=xxxxxxxxxxxxxxxx*]

**Diagram**




### *7.24.3 Parameter*

**nn** This specifies the optional register number (00-15) to be altered.

**xxxxxxxx** This is the register value in hexadecimal (1-8 hex digits for 32-bit registers, 1-16 hex digits for 64-bit registers).

### *7.24.4 Examples*

**Example 1:**

Display the control registers.

```
14:22:33.559 00000FC4 cr
14:22:33.559 00000FC4 CPU0000: CR00=5FB1EE40  CR01=3FFFE07F  CR02=3F149FC0  CR03=00000001
14:22:33.559 00000FC4 CPU0000: CR04=00010001  CR05=3FFE7040  CR06=FE000000  CR07=3FFFE07F
14:22:33.559 00000FC4 CPU0000: CR08=00000000  CR09=00000000  CR10=00000000  CR11=00000000
14:22:33.559 00000FC4 CPU0000: CR12=01984E80  CR13=3FFFE07F  CR14=FF8BF15F  CR15=03150010
```

**Figure 46: CR command (display control registers)**

**Example 2:**

Alter control register number 15.

```
14:22:33.559 00000FC4 cr 15=FFFFFFFF
14:22:33.559 00000FC4 CPU0000: CR00=5FB1EE40  CR01=3FFFE07F  CR02=3F149FC0  CR03=00000001
14:22:33.559 00000FC4 CPU0000: CR04=00010001  CR05=3FFE7040  CR06=FE000000  CR07=3FFFE07F
14:22:33.559 00000FC4 CPU0000: CR08=00000000  CR09=00000000  CR10=00000000  CR11=00000000
14:22:33.559 00000FC4 CPU0000: CR12=01984E80  CR13=3FFFE07F  CR14=FF8BF15F  CR15=FFFFFFFF
```

**Figure 47: CR command (alter control register)**

## 7.25 CSCRIPT (Cancel a running script thread)

### *7.25.1 Function*

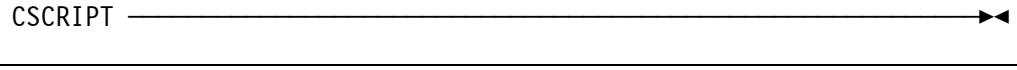
The CSCRIPT command will cancel the currently running script. If no script is running, no action is taken.

### *7.25.2 Syntax*

**Descriptive**

```
CSCRIPT
```

**Diagram**




### *7.25.3 Parameter*

None.

### *7.25.4 Examples*

**Example 1:**

Cancel the currently running script.

```
2007/06/30 13:13:05.656 00000FC4 cscript
```

**Figure 48: CSCRIPT command**

## 7.26 CTC (Enable / disable debug packet tracing)

### *7.26.1 Function*

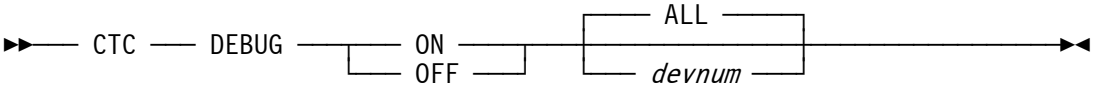
The CTC command enables or disables the debug packet tracing for the specified CTCI/LCS device group(s) identified by *devnum* or for all CTCI/LCS device group(s) if *devnum* is not specified or specified as ALL.

### *7.26.2 Syntax*

**Descriptive**

```
CTC DEBUG {ON | OFF} [devnum | ALL]
```

**Diagram**




### *7.26.3 Parameter*

**ON** Enables the debug packet tracing.

**OFF** Disables the debug packet tracing.

**devnum** Specifies the CTCI/LCS device group(s) for which debug packet tracing has to be enabled or disabled.

**ALL** Enables or disables the debug packet tracing for all CTCI/LCS device groups. 'ALL' is the default if no device group is specified.

### *7.26.4 Examples*

**Example 1:**

Enable the debug packet tracing for all CTCI/LCS devices.

```
2008/01/27 19:04:11.375 00000378 ctc debug on
2008/01/27 19:04:11.375 00000378 HHCPNXXXI CTC debugging now ON for all CTCI/LCS device groups.
2008/01/27 19:04:30.906 00000378 HHCCT046I 0E20: Sending packet to tun0:
2008/01/27 19:04:30.906 00000378 +0000  4500011C 00AC0000 4001F71C C0A80063  E.......@.7.{y.c
2008/01/27 19:04:30.937 00000378 +0010  C0A80065 0800E930 93363000 C1DC84EA  {y.e..Z0l60.A.d.
2008/01/27 19:04:30.937 00000378 +0020  928D5000 031181A7 7256FAF1 2C74E437  k.P...axrV.1,tU7
```

```
2008/01/27 19:04:30.937 00000378 +0030  2E32D6E1 0BB94DC7 32A143D1 796B4E57  .2O...MG2~CJykNW
2008/01/27 19:04:30.937 00000378 +0040  73E431C1 0A7E75E7 746E90B1 06DD5477  sU1A.~uXtn....Tw
2008/01/27 19:04:30.937 00000378 +0050  2CD450A1 26F97A07 44BA6191 285B7697  ,MP~&9z.D.aj([vp
2008/01/27 19:04:30.937 00000378 +0060  0CC6B381 4932DA27 29903671 3AE634B7  .F.aI2.').6q:W4.
2008/01/27 19:04:30.937 00000378 +0070  5F8EDA61 1BA31647 700B8F51 06EE0ED7  _..a.t.Gp..Q...P
2008/01/27 19:04:30.937 00000378 +0080  77104541 4D32AE67 3A57EC31 625384F7  w.EAM2.g:W.1bSd7
2008/01/27 19:04:30.937 00000378 +0090  593E7421 133A2287 4DB0CD11 12671717  Y>t!.:"gM....g..
2008/01/27 19:04:30.937 00000378 +00A0  2E1CE701 6B81F2A7 2061B1F1 6DE94537  ..X.ka2x a.1mZE7
2008/01/27 19:04:30.937 00000378 +00B0  6DBF1DE1 1E429EC7 27C61AD1 3F0A8F57  m....B.G'F.J?..W
2008/01/27 19:04:30.937 00000378 +00C0  4E4898C1 0024A6E7 664987B1 656B7577  NHqA.$wXfIg.ekuw
2008/01/27 19:04:30.937 00000378 +00D0  71ECD7A1 74408B07 39677891 381C7797  q.P~t@..9gxj8.wp
2008/01/27 19:04:30.937 00000378 +00E0  54EF5A81 2E1ECB27 67AB6D71 279E15B7  T.Za...'g.mq'...
2008/01/27 19:04:30.937 00000378 +00F0  7BA3A161 33B7E747 6EB0E651 1FE0CFD7  {t~a3.XGn.WQ.\.P
2008/01/27 19:04:30.937 00000378 +0100  606D2C41 1F745F67 11236331 2A4525F7  `m,A.t_g.#c1*E%7
2008/01/27 19:04:30.937 00000378 +0110  21BF7B21 222CB387 24BE6411           !.{!",.g$.d.
2008/01/27 19:04:30.937 00000378 HHCCT049I 0E20: Received packet from tun0 (284 bytes):
2008/01/27 19:04:30.937 00000378 +0000  4500011C A2570000 80011571 C0A80065  E...sW.....q{y.e
2008/01/27 19:04:30.937 00000378 +0010  C0A80063 0000F130 93363000 C1DC84EA  {y.c..10l60.A.d.
2008/01/27 19:04:30.937 00000378 +0020  928D5000 031181A7 7256FAF1 2C74E437  k.P...axrV.1,tU7
2008/01/27 19:04:30.937 00000378 +0030  2E32D6E1 0BB94DC7 32A143D1 796B4E57  .2O...MG2~CJykNW
2008/01/27 19:04:30.937 00000378 +0040  73E431C1 0A7E75E7 746E90B1 06DD5477  sU1A.~uXtn....Tw
2008/01/27 19:04:30.937 00000378 +0050  2CD450A1 26F97A07 44BA6191 285B7697  ,MP~&9z.D.aj([vp
2008/01/27 19:04:30.937 00000378 +0060  0CC6B381 4932DA27 29903671 3AE634B7  .F.aI2.').6q:W4.
2008/01/27 19:04:30.937 00000378 +0070  5F8EDA61 1BA31647 700B8F51 06EE0ED7  _..a.t.Gp..Q...P
2008/01/27 19:04:30.937 00000378 +0080  77104541 4D32AE67 3A57EC31 625384F7  w.EAM2.g:W.1bSd7
2008/01/27 19:04:30.937 00000378 +0090  593E7421 133A2287 4DB0CD11 12671717  Y>t!.:"gM....g..
2008/01/27 19:04:30.937 00000378 +00A0  2E1CE701 6B81F2A7 2061B1F1 6DE94537  ..X.ka2x a.1mZE7
2008/01/27 19:04:30.937 00000378 +00B0  6DBF1DE1 1E429EC7 27C61AD1 3F0A8F57  m....B.G'F.J?..W
2008/01/27 19:04:30.937 00000378 +00C0  4E4898C1 0024A6E7 664987B1 656B7577  NHqA.$wXfIg.ekuw
2008/01/27 19:04:30.937 00000378 +00D0  71ECD7A1 74408B07 39677891 381C7797  q.P~t@..9gxj8.wp
2008/01/27 19:04:30.937 00000378 +00E0  54EF5A81 2E1ECB27 67AB6D71 279E15B7  T.Za...'g.mq'...
2008/01/27 19:04:30.937 00000378 +00F0  7BA3A161 33B7E747 6EB0E651 1FE0CFD7  {t~a3.XGn.WQ.\.P
2008/01/27 19:04:30.937 00000378 +0100  606D2C41 1F745F67 11236331 2A4525F7  `m,A.t_g.#c1*E%7
2008/01/27 19:04:30.937 00000378 +0110  21BF7B21 222CB387 24BE6411           !.{!",.g$.d.
2008/01/27 19:04:30.937 00000378 HHCCT041I 0E21: CTC Received Frame (292 bytes):
2008/01/27 19:04:30.937 00000378 +0000  01240122 08000000 4500011C A2570000  .$."....E...sW..
2008/01/27 19:04:30.937 00000378 +0010  80011571 C0A80065 C0A80063 0000F130  ...q{y.e{y.c..10
2008/01/27 19:04:30.937 00000378 +0020  93363000 C1DC84EA 928D5000 031181A7  l60.A.d.k.P...ax
2008/01/27 19:04:30.937 00000378 +0030  7256FAF1 2C74E437 2E32D6E1 0BB94DC7  rV.1,tU7.2O...MG
2008/01/27 19:04:30.937 00000378 +0040  32A143D1 796B4E57 73E431C1 0A7E75E7  2~CJykNWsU1A.~uX
2008/01/27 19:04:30.937 00000378 +0050  746E90B1 06DD5477 2CD450A1 26F97A07  tn....Tw,MP~&9z.
2008/01/27 19:04:30.937 00000378 +0060  44BA6191 285B7697 0CC6B381 4932DA27  D.aj([vp.F.aI2.'
```

```
2008/01/27 19:04:30.937 00000378 +0070  29903671 3AE634B7 5F8EDA61 1BA31647  ).6q:W4._..a.t.G
2008/01/27 19:04:30.937 00000378 +0080  700B8F51 06EE0ED7 77104541 4D32AE67  p..Q...Pw.EAM2.g
2008/01/27 19:04:30.937 00000378 +0090  3A57EC31 625384F7 593E7421 133A2287  :W.1bSd7Y>t!.:"g
2008/01/27 19:04:30.937 00000378 +00A0  4DB0CD11 12671717 2E1CE701 6B81F2A7  M....g....X.ka2x
2008/01/27 19:04:30.937 00000378 +00B0  2061B1F1 6DE94537 6DBF1DE1 1E429EC7   a.1mZE7m....B.G
2008/01/27 19:04:30.937 00000378 +00C0  27C61AD1 3F0A8F57 4E4898C1 0024A6E7  'F.J?..WNHqA.$wX
2008/01/27 19:04:30.937 00000378 +00D0  664987B1 656B7577 71ECD7A1 74408B07  fIg.ekuwq.P~t@..
2008/01/27 19:04:30.937 00000378 +00E0  39677891 381C7797 54EF5A81 2E1ECB27  9gxj8.wpT.Za...'
2008/01/27 19:04:30.937 00000378 +00F0  67AB6D71 279E15B7 7BA3A161 33B7E747  g.mq'...{t~a3.XG
2008/01/27 19:04:30.937 00000378 +0100  6EB0E651 1FE0CFD7 606D2C41 1F745F67  n.WQ.\.P`m,A.t_g
2008/01/27 19:04:30.937 00000378 +0110  11236331 2A4525F7 21BF7B21 222CB387  .#c1*E%7!.{!",.g
2008/01/27 19:04:30.937 00000378 +0120  24BE6411                           $.d.
2008/01/27 19:04:44.406 00000378 HHCCT049I 0E20: Received packet from tun0 (229 bytes):
2008/01/27 19:04:44.421 00000378 +0000  450000E5 02180000 8011B53A C0A80066  E..V.......:{y.f
2008/01/27 19:04:44.421 00000378 +0010  C0A800FF 008A008A 00D1B65E 1102804F  {y.......J.^...O
2008/01/27 19:04:44.421 00000378 +0020  C0A80066 008A00BB 00002045 4D454246  {y.f...... EMEBF
2008/01/27 19:04:44.421 00000378 +0030  43454243 41434143 41434143 41434143  CEBCACACACACACAC
2008/01/27 19:04:44.421 00000378 +0040  41434143 41434143 41434100 20464445  ACACACACACA. FDE
2008/01/27 19:04:44.421 00000378 +0050  50464345 44454646 43454646 43434143  PFCEDEFFCEFFCCAC
2008/01/27 19:04:44.421 00000378 +0060  41434143 41434143 41434142 4E00FF53  ACACACACACABN..S
2008/01/27 19:04:44.421 00000378 +0070  4D422500 00000000 00000000 00000000  MB%.............
2008/01/27 19:04:44.421 00000378 +0080  00000000 00000000 00000000 00001100  ................
2008/01/27 19:04:44.421 00000378 +0090  00210000 00000000 000000E8 03000000  .!.........Y....
2008/01/27 19:04:44.421 00000378 +00A0  00000000 00210056 00030001 00000002  .....!.V........
2008/01/27 19:04:44.421 00000378 +00B0  0032005C 4D41494C 534C4F54 5C42524F  .2.\MAILSLOT\BRO
2008/01/27 19:04:44.421 00000378 +00C0  57534500 010080FC 0A004C41 52410000  WSE.......LARA..
2008/01/27 19:04:44.421 00000378 +00D0  00000000 24000000 00000501 02100000  ....$...........
2008/01/27 19:04:44.421 00000378 +00E0  0F0155AA 00                          ..U..
.
.
.
```

**Figure 49: CTC command**

## 7.27 DEFINE (Rename device)

### *7.27.1 Function*

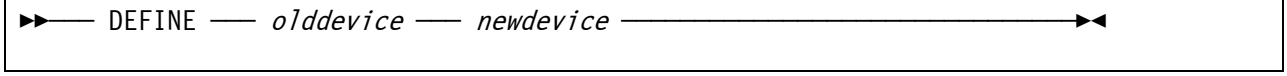
The DEFINE command can be used to “rename” a device (change the device number).

### *7.27.2 Syntax*

**Descriptive**

DEFINE *olddevice newdevice*

**Diagram**




### *7.27.3 Parameter*

**olddevice** The device number of the device that has to be renamed.

**newdevice** The new device number, to which the device should now be attached.

### *7.27.4 Examples*

**Example 1:**

Rename two devices.

```
2007/06/30 14:31:59.718 00000FC4 define 0704 0705
2007/06/30 14:32:20.328 00000FC4 define 0a8c 0a8d
```

**Figure 50: DEFINE command**

## 7.28 DEFSYM (Define symbol)

### 7.28.1 Function

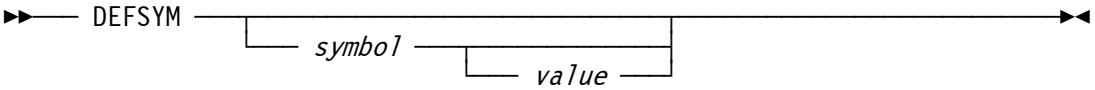
The DEFSYM command defines symbol *symbol* as to contain value *value*. The symbol can be the object of a substitution. If the *value* contains blanks or spaces then it must be enclosed within quotes or apostrophes. See chapter “Symbol Substitutions” for a more in-depth discussion on this feature.

### 7.28.2 Syntax

**Descriptive**

DEFSYM [*symbol* [*value*]]

**Diagram**




### 7.28.3 Parameter

**symbol** The name of a symbol. If no symbol name (and optionally a value) is given then all previously defined symbols are listed.

**value** The value that is assigned to the symbol. If the value is omitted then the previously defined symbol is cleared, the symbol itself is not deleted.

### 7.28.4 Examples

**Example 1:**

List all defined symbols.

```
2007/06/30 12:44:06.890 00000FC4 defsym
2007/06/30 12:44:06.890 00000FC4 HHCPN042I CUU=$(CUU)
2007/06/30 12:44:06.890 00000FC4 HHCPN042I cuu=$(cuu)
2007/06/30 12:44:06.890 00000FC4 HHCPN042I CCUU=$(CCUU)
2007/06/30 12:44:06.890 00000FC4 HHCPN042I ccuu=$(ccuu)
2007/06/30 12:44:06.890 00000FC4 HHCPN042I DASDPATH=D:/MVS/DASD_CCKD
2007/06/30 12:44:06.890 00000FC4 HHCPN042I CSS=0
```

**Figure 51: DEFSYM command (list all symbols)**

**Example 2:**

Define new symbol (and list the symbols afterwards).

```
2007/06/30 12:44:36.187 00000ED0 defsym tapedir "/home/hercules/tapes"
2007/06/30 12:44:42.296 00000ED0 defsym
2007/06/30 12:44:42.296 00000ED0 HHCPN042I CUU=$(CUU)
2007/06/30 12:44:42.296 00000ED0 HHCPN042I cuu=$(cuu)
2007/06/30 12:44:42.296 00000ED0 HHCPN042I CCUU=$(CCUU)
2007/06/30 12:44:42.296 00000ED0 HHCPN042I ccuu=$(ccuu)
2007/06/30 12:44:42.296 00000ED0 HHCPN042I CSS=0
2007/06/30 12:44:42.296 00000ED0 HHCPN042I tapedir="/home/hercules/tapes"
```

**Figure 52: DEFSYM command (define new symbol)**

**Example 3:**

Clear defined symbol.

```
2007/06/30 12:44:49.281 00000ED0 defsym tapedir
2007/06/30 12:44:52.625 00000ED0 defsym
2007/06/30 12:44:52.625 00000ED0 HHCPN042I CUU=$(CUU)
2007/06/30 12:44:52.625 00000ED0 HHCPN042I cuu=$(cuu)
2007/06/30 12:44:52.625 00000ED0 HHCPN042I CCUU=$(CCUU)
2007/06/30 12:44:52.625 00000ED0 HHCPN042I ccuu=$(ccuu)
2007/06/30 12:44:52.625 00000ED0 HHCPN042I CSS=0
2007/06/30 12:44:52.625 00000ED0 HHCPN042I tapedir=
```

**Figure 53: DEFSYM command (clear defined symbol)**

## 7.29 DETACH (Remove device)

### *7.29.1 Function*

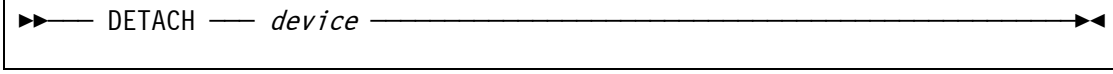
The DETACH command removes a device from the current configuration. The effect of this command is immediately visible in the device display of the Hercules Windows GUI and the Hercules “Peripherals” display.

### *7.29.2 Syntax*

**Descriptive**

```
DETACH device
```

**Diagram**




### *7.29.3 Parameter*

**device** The device number of the device that is to be removed from the current configuration.

### *7.29.4 Examples*

**Example 1:**

Detach device 000F from the active configuration.

```
2007/06/30 13:31:32.187 00000FC4 detach 000f
2007/06/30 13:31:32.187 00000FC4 HHCCF047I Device 000F detached
```

**Figure 54: DETACH command**

## 7.30 DEVINIT (Reinitialize device)

### *7.30.1 Function*

The DEVINIT command can be used to reopen an existing device. The arguments (if any) override the arguments specified in the configuration file for this device. If no arguments (beneath the device number) are given, then the DEVINIT command will reinitialise the device with the same arguments that were previously specified.The device type cannot be changed and must not be specified.

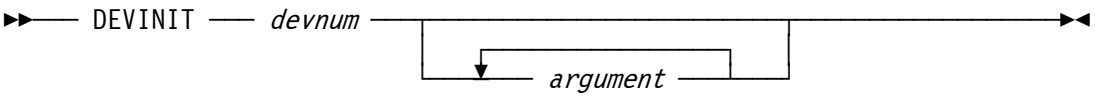
This command can be used to rewind a tape, to mount a new tape or disk image file on an existing device, to load a new card deck into a reader, or to close and reopen a printer or punch device.

### *7.30.2 Syntax*

**Descriptive**

DEVINIT *devnum* [*argument* [*argument* ... ]]

**Diagram**




### *7.30.3 Parameter*

**devnum** The device number of the device to be reinitialized.

**argument** Any arguments that are acceptable for the kind of device being reinitalized.

### *7.30.4 Examples*

**Example 1:**

Initialize a tape device.

```
00000FC4 devinit 0561 D:/MVS/TAPE/TST001.HET chunksize=65535 compress=1 method=1 level=4
00000FC4 HHCTA998I Device 0561 : D:/MVS/TAPE/TST001.HET is a Hercules Formatted Tape
00000FC4 XXXXXXXXX Device 0561 : parameter: 'chunksize=65535'
00000FC4 XXXXXXXXX Device 0561 : parameter: 'compress=1'
00000FC4 XXXXXXXXX Device 0561 : parameter: 'method=1'
00000FC4 XXXXXXXXX Device 0561 : parameter: 'level=4'
00000FC4 HHCTA100I 0561: Now Displays: " NT RDY "
00000FC4 HHCPN098I Device 0561 initialized
```

**Figure 55: DEVINIT command**

## 7.31 DEVLIST (List device or all devices)

### *7.31.1 Function*

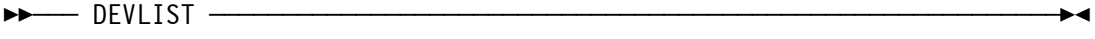
The DEVLIST command lists all the devices that are defined in the active Hercules configuration and displays their current status.

### *7.31.2 Syntax*

**Descriptive**

```
DEVLIST
```

**Diagram**




### *7.31.3 Parameter*

None.

### *7.31.4 Examples*

**Example 1:**

List all devices of the current configuration.

```
2007/05/30 23:07:22.181 00000F28 devlist
2007/05/30 23:07:22.197 00000F28 0:000C 3505 D://MVS/JCL/DUMMY.JCL ascii eof open
2007/05/30 23:07:22.197 00000F28 0:000D 3525 D://MVS/PCH/pch00d.txt ascii crlf open
2007/05/30 23:07:22.197 00000F28 0:000E 1403 D://MVS/PRT/prt00e.txt crlf open
2007/05/30 23:07:22.197 00000F28 0:000F 1403 D://MVS/PRT/prt00f.txt crlf open
2007/05/30 23:07:22.197 00000F28 0:0010 3270 192.168.0.101 open
2007/05/30 23:07:22.197 00000F28 0:0011 3270 192.168.0.101 open
2007/05/30 23:07:22.197 00000F28 0:00C0 3270 192.168.0.101 open
2007/05/30 23:07:22.197 00000F28 0:00C1 3270 192.168.0.101 open
2007/05/30 23:07:22.197 00000F28 0:00C2 3270 192.168.0.101 open
2007/05/30 23:07:22.197 00000F28 0:00C3 3270
2007/05/30 23:07:22.197 00000F28 0:00C4 3270
2007/05/30 23:07:22.197 00000F28 0:00C5 3270
2007/05/30 23:07:22.197 00000F28 0:00C6 3270
2007/05/30 23:07:22.197 00000F28 0:00C7 3270
2007/05/30 23:07:22.197 00000F28 0:0130 2314 D://MVS/DASD/SORT00.CCKD [200 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0131 2314 D://MVS/DASD/SORT01.CCKD [200 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0132 2314 D://MVS/DASD/SORT02.CCKD [203 cyls] open
```

```
2007/05/30 23:07:22.197 00000F28 0:0133 2314 D://MVS/DASD/SORT03.CCKD [200 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0134 2314 D://MVS/DASD/SORT04.CCKD [200 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0135 2314 D://MVS/DASD/SORT05.CCKD [200 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0140 3350 D://MVS/DASD/WORK00.CCKD [555 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0141 3350 D://MVS/DASD/WORK01.CCKD [555 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0142 3350 D://MVS/DASD/PRD000.CCKD [560 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0143 3350 D://MVS/DASD/PRD001.CCKD [560 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0144 3350 D://MVS/DASD/PRD002.CCKD [560 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0145 3350 D://MVS/DASD/PRD003.CCKD [560 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0148 3350 D://MVS/DASD/MVSRES.CCKD [560 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0149 3350 D://MVS/DASD/MVSDLB.CCKD [555 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0160 3340 D://MVS/DASD/PAGE00.CCKD [698 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0161 3340 D://MVS/DASD/PAGE01.CCKD [698 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0240 3350 D://MVS/DASD/PUB000.CCKD [555 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0241 3350 D://MVS/DASD/PUB001.CCKD [555 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0242 3350 D://MVS/DASD/PUB002.CCKD [560 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0243 3350 D://MVS/DASD/PUB003.CCKD [560 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0248 3350 D://MVS/DASD/SYS000.CCKD [560 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0249 3350 D://MVS/DASD/SYS001.CCKD [560 cyls] open
2007/05/30 23:07:22.197 00000F28 0:024A 3350 D://MVS/DASD/SYS002.CCKD [560 cyls] open
2007/05/30 23:07:22.197 00000F28 0:024B 3350 D://MVS/DASD/SYS003.CCKD [560 cyls] open
2007/05/30 23:07:22.197 00000F28 0:030E 1403 D://MVS/LOG/hardcopy.log crlf open
2007/05/30 23:07:22.197 00000F28 0:0340 3350 D://MVS/DASD/CBT000.CCKD [555 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0341 3350 D://MVS/DASD/CBT001.CCKD [555 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0342 3350 D://MVS/DASD/CBT002.CCKD [555 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0343 3350 D://MVS/DASD/CBT003.CCKD [555 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0344 3350 D://MVS/DASD/SPOOL0.CCKD [560 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0345 3350 D://MVS/DASD/SPOOL1.CCKD [560 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0348 3350 D://MVS/DASD/TST000.CCKD [560 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0349 3350 D://MVS/DASD/TST001.CCKD [560 cyls] open
2007/05/30 23:07:22.197 00000F28 0:034A 3350 D://MVS/DASD/TST002.CCKD [560 cyls] open
2007/05/30 23:07:22.197 00000F28 0:034B 3350 D://MVS/DASD/TST003.CCKD [560 cyls] open
2007/05/30 23:07:22.197 00000F28 0:0480 3420 *
```

**Figure 56: DEVLIST command**

## 7.32 DEVTMAX (Display or set maximum device threads)

### *7.32.1 Function*

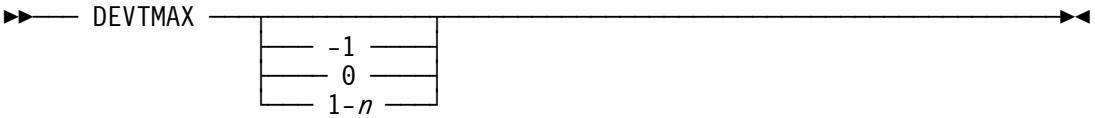
The DEVTMAX command displays the current setting of the DEVTMAX system parameter along with the actual device threads statistics. As an additional argument a new value for the maximum device threads may be specified.

### *7.32.2 Syntax*

**Descriptive**

```
DEVTMAX [-1 | 0 | 1-n]
```

**Diagram**




### *7.32.3 Parameter*

**-1** Specifiy -1 to cause ‘one time only’ temporary threads to be created to service each I/O request to a device. Once the I/O request is complete the thread exits. Subsequent I/Os to the same device will cause another worker thread to be created again.

**0** Specify 0 to cause an unlimited number of ‘semi-permanent’ threads to be created on an ‘as-needed’ basis. With this option a thread is created to service an I/O request for a device if one does not already exist. When the I/O is complete then the thread enters an idle state waiting for new work. If a new I/O request for the device arrives before the timeout period expires, the existing thread will be reused. The timeout value is currently hard coded at 5 minutes.

Note that this option can cause one thread (or possibly more) to be created for each device in your configuration. Specifying 0 means there is no limit to the number of threads that can be created.

**1-n** Specifiy a value from 1 to n to set an upper limit to the number of threads that can be created to service any I/O request to any device. Like the “0” option, each thread, once done servicing an I/O request enters an idle state. If a new request arrives before the timeout period expires the thread is reused. If all threads are busy when a new I/O request arrives, a new thread is created only if the specified maximum number of threads have not yet been reached. If the specified maximum number of threads already has been reached then the I/O request is placed in a queue and will be serviced by the first available thread (i.e. whichever thread becomes idle first).

This option was created to address a threading issue, possibly related to the Cygwin phtreads implementation on Windows systems. The default for Windows is 8. The default for all other systems is 0.

## *7.32.4 Examples*

**Example 1:**

Display the maximum allowed device threads along with the current statistics.

```
00000FC4 devtmax
00000FC4 HHCPN076I Max device threads: 0, current: 2, most: 5, waiting: 0, max exceeded: 0
```

**Figure 57: DEVTMAX command (list maximum device threads)**

**Example 2:**

Set a new value for the maximum allowed device threads.

```
00000FC4 devtmax 25
00000FC4 HHCPN076I Max device threads: 25, current: 2, most: 5, waiting: 0, max exceeded: 0
```

**Figure 58: DEVTMAX command (set maximum device threads)**

## 7.33 DS (Display subchannel)

### 7.33.1 Function

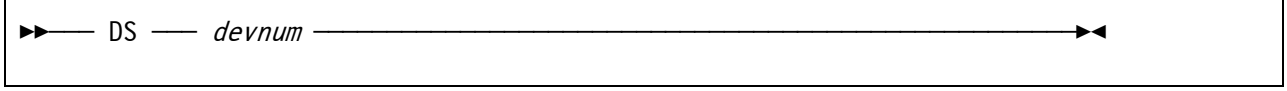
The DS command shows the subchannel information for a given device number.

### 7.33.2 Syntax

**Descriptive**

DS *devnum*

**Diagram**




### 7.33.3 Parameter

**devnum** The device number, for which the subchannel information has to be displayed.

### 7.33.4 Examples

**Example 1:**

Display subchannel information for device 0AE0.

```
000009EC ds 0ae0
000009EC 0AE0:D/T=3390 Subchannel_Number=0028
000009EC      PMCW=IntParm:00F3F8A8 Flags:1899 Dev:0AE0 LPM:80 PNOM:00 LPUM:80 PIM:80
000009EC           MBI:0183 POM:FF PAM:80 CHPIDs:0A00000000000000 Misc:00000001
000009EC      SCSW=Flags:08C0 SCHC:4020 Stat:0000 Count:0001 CCW:3727CCF0
```

**Figure 59: DS command**

### 7.33.5 Explanations

The information presented in the output from the Display Subchannel command is taken from the Sub-channel-Information Block (SCHIB). The Subchannel-Information Block shows - beneath other areas - the Path-Management-Control Word (PMCW) and the Subchannel-Status Word (SCSW). The format of the SCHIB is shown on the following page.

| Word | |
|---|---|
| 0 | Path-Management-Control Word |
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| 5 | |
| 6 | |
| 7 | Subchannel-Status Word |
| 8 | |
| 9 | |
| 10 | Model-Dependent Area / |
| 11 | Measurement-Block Adress |
| 12 | Model-Dependent Area |

**Figure 60: Subchannel-Information Block**

Words 0-6 of the SCHIB contain the Path-Management-Control Word (PMCW). The format of the PMCW is the following:

| Word | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Interruption Parameter | | | | | | | | | | | | | | | | | | |
| 1 | 00 | ISC | | 000 | | E | LM | MM | D | T | V | Device Number | | | | | | | |
| 2 | LPM | | | | | PNOM | | | | | | LPUM | | | | PIM | | | |
| 3 | MBI | | | | | | | | | | | POM | | | | PAM | | | |
| 4 | CHPID-0 | | | | | CHPID-1 | | | | | | CHPID-2 | | | | CHPID-3 | | | |
| 5 | CHPID-4 | | | | | CHPID-5 | | | | | | CHPID-6 | | | | CHPID-7 | | | |
| 6 | 0 0 0 0 0 0 0 0 | | | | | 0 0 0 0 0 0 0 0 | | | | | | 0 0 0 0 0 0 0 0 | | | | 0 | 0 | 0 | 0 | 0 | F | X | S |

**Figure 61: Path-Management-Control Word**

Legend:

| | |
|---|---|
| ISC | Interruption-Subclass Code |
| E | Enabled |
| LM | Limit Mode |
| MM | Measurement-Mode Enable |
| D | Multipath Mode |
| T | Timing Facility |
| V | Device Number Valid |
| LPM | Logical-Path Mask |
| PNOM | Path-Not-Operational Mask |
| LPUM | Last-Path-Used Mask |
| PIM | Path-Installed Mask |
| MBI | Measurement-Block Index |
| POM | Path-Operational Mask |

| | |
|---|---|
| PAM | Path-Available Mask |
| CHPIDs | Channel-Path Identifiers |
| F | Measurement Block Format Control |
| X | Extended Measurement Word Enable |
| S | Concurrent Sense |

The Subchannel-Status Word (SCSW) provides information about the status of a subchannel and its associated devices. The format of the SCSW is as follows:

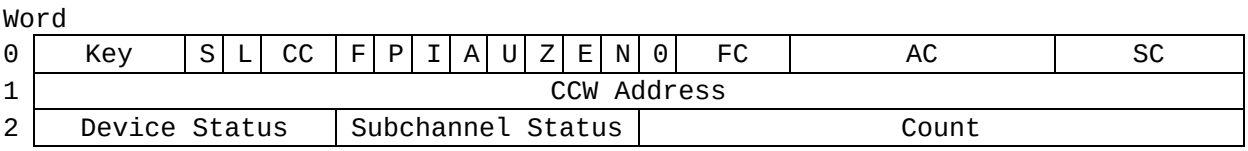



**Figure 62: Subchannel-Status Word**

Legend:

| | |
|---|---|
| Key | Subchannel Key |
| S | Suspend Control |
| L | ESW Format |
| CC | Deferred Condition Code |
| F | Format |
| P | Prefetch |
| I | Initial-Status Interruption Control |
| A | Address-Limit-Checking Control |
| U | Suppress-Suspend Interruption |
| Z | Zero Condition Code |
| E | Extended Control |
| N | Path Not Operational |
| 0 | Reserved |
| FC | Function Control |
| AC | Activity Control |
| SC | Status Control |
| Count | Residual Count |

Details about the control blocks, that have been described here only briefly, can be found in IBM's "z/Architecture Principles of Operation" manual.

## 7.34 ECPSVM (ECPS:VM commands)

### *7.34.1 Function*

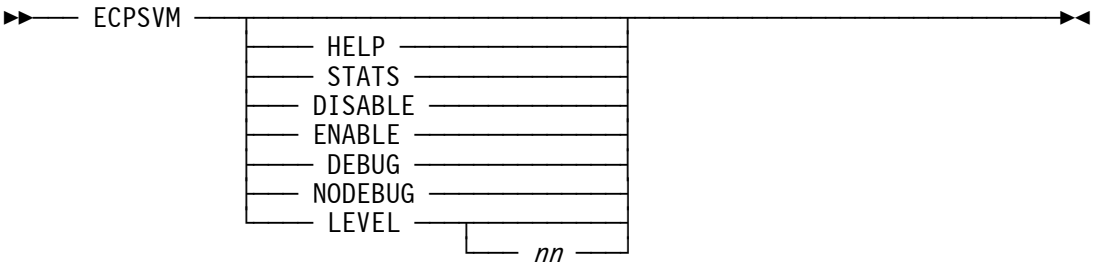
The ECPSVM command invokes ECPS:VM subcommands (described below). Issuing command ECPSVM HELP displays a list of all available commands.

### *7.34.2 Syntax*

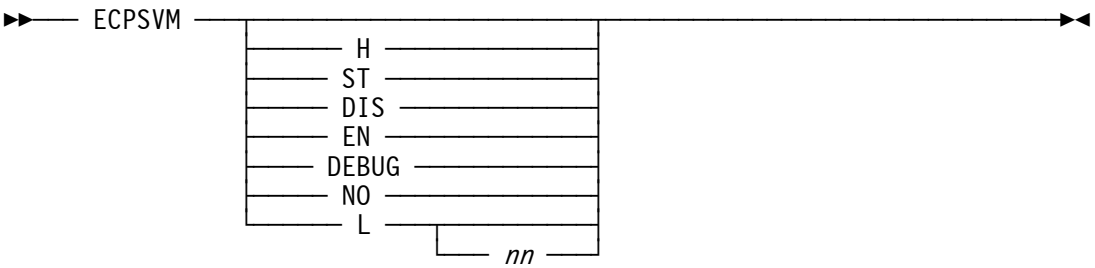
**Descriptive**

```
ECPSVM [HELP | STATS | DISABLE | ENABLE | DEBUG | NODEBUG | LEVEL [nn]]

  or (with abbreviated arguments)

ECPSVM [H | ST | DIS | EN | DEBUG | NO | L [nn]]
```

**Diagram**




or (with abbreviated arguments)



## 7.34.3 Parameter

**HELP** Show the help panel with all available commands.

**STATS** Show statistical counters.

**DISABLE** Disable all ECPS:VM features.

**ENABLE** Enable all ECPS:VM features.

**DEBUG** Debug ECPS:VM features.

**NODEBUG** Turn debug modus off for ECPS:VM features.

**LEVEL** Set/show ECPS:VM level.

**nn** The value nn specifies the support level that is reported to the operating system.

## 7.34.4 Examples

**Example 1:**

Show ECPS:VM statistical counters.

```
2007/06/30 17:16:05.656 00000FC4 ecpsvm stats
2007/06/30 17:16:05.656 00000FC4 HHCEV011I ECPS:VM Command processor invoked
2007/06/30 17:16:05.656 00000FC4 HHCEV003I +-----------+----------+----------+-------+
2007/06/30 17:16:05.656 00000FC4 HHCEV002I | VM ASSIST | Calls    | Hits     | Ratio |
2007/06/30 17:16:05.656 00000FC4 HHCEV003I +-----------+----------+----------+-------+
2007/06/30 17:16:05.656 00000FC4 HHCEV001I | Total     |        0 |        0 |  100% |
2007/06/30 17:16:05.656 00000FC4 HHCEV003I +-----------+----------+----------+-------+
2007/06/30 17:16:05.656 00000FC4 HHCEV005I 11 Entries not shown (never invoked)
2007/06/30 17:16:05.656 00000FC4 HHCEV003I +-----------+----------+----------+-------+
2007/06/30 17:16:05.656 00000FC4 HHCEV002I | CP ASSIST | Calls    | Hits     | Ratio |
2007/06/30 17:16:05.656 00000FC4 HHCEV003I +-----------+----------+----------+-------+
2007/06/30 17:16:05.656 00000FC4 HHCEV001I | Total     |        0 |        0 |  100% |
2007/06/30 17:16:05.656 00000FC4 HHCEV003I +-----------+----------+----------+-------+
2007/06/30 17:16:05.656 00000FC4 HHCEV005I 23 Entries not shown (never invoked)
2007/06/30 17:16:05.656 00000FC4 HHCEV011I ECPS:VM Command processor complete
```

**Figure 63: ECPSVM STATS command**

**Example 2:**

Enable the ECPS:VM features.

```
2007/06/30 17:16:55.468 00000FC4 ecpsvm enable
2007/06/30 17:16:55.468 00000FC4 HHCEV011I ECPS:VM Command processor invoked
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM VM ASSIST feature SVC Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM VM ASSIST feature SSM Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM VM ASSIST feature LPSW Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM VM ASSIST feature STNSM Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM VM ASSIST feature STOSM Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM VM ASSIST feature SIO Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM VM ASSIST feature VTIMER Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM VM ASSIST feature STCTL Enabled
```

```
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM VM ASSIST feature LCTL Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM VM ASSIST feature DIAG Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM VM ASSIST feature IUCV Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV016I All ECPS:VM VM ASSIST features Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature FREE Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature FRET Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature LCKPG Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature ULKPG Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature SCNRU Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature SCNVU Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature DISP0 Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature DISP1 Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature DISP2 Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature DNCCW Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature DFCCW Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature FCCWS Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature CCWGN Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature UXCCW Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature TRBRG Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature TRLOK Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature VIST Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature VIPT Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature STEVL Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature FREEX Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature FRETX Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature PMASS Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV015I ECPS:VM CP ASSIST feature LCSPG Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV016I All ECPS:VM CP ASSIST features Enabled
2007/06/30 17:16:55.468 00000FC4 HHCEV011I ECPS:VM Command processor complete
```

**Figure 64: ECPSVM ENABLE command**

**Example 3:**

Show the ECPS:VM level.

```
2007/06/30 17:16:40.843 00000FC4 ecpsvm level
2007/06/30 17:16:40.843 00000FC4 HHCEV011I ECPS:VM Command processor invoked
2007/06/30 17:16:40.843 00000FC4 HHCEV016I Current reported ECPS:VM Level is 0
2007/06/30 17:16:40.843 00000FC4 HHCEV017I But ECPS:VM is currently disabled
2007/06/30 17:16:40.843 00000FC4 HHCEV017W WARNING ! current level (0) is not supported
2007/06/30 17:16:40.843 00000FC4 HHCEV018W WARNING ! Unpredictable results may occur
2007/06/30 17:16:40.843 00000FC4 HHCEV019I The microcode support level is 20
2007/06/30 17:16:40.843 00000FC4 HHCEV011I ECPS:VM Command processor complete
```

**Figure 65: ECPSVM LEVEL command**

## 7.35 EXIT (Terminate the emulator)

### *7.35.1 Function*

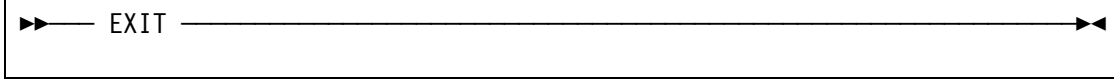
The EXIT command (see also the QUIT command) initiates the Hercules shutdown. It terminates all threads, detaches all channels and devices and releases the configuration. Finally it terminates the emulator.

### *7.35.2 Syntax*

**Descriptive**

```
EXIT
```

**Diagram**




### *7.35.3 Parameter*

None.

### *7.35.4 Examples*

**Example 1:**

Initiate the Hercules shutdown.

```
2007/06/30 2007/06/30 14:49:57.044 00000FC4 exit
2007/06/30 2007/06/30 14:49:57.044 00000FC4 HHCIN900I Begin Hercules shutdown
2007/06/30 2007/06/30 14:49:57.044 00000FC4 HHCIN901I Releasing configuration
2007/06/30 2007/06/30 14:49:57.044 00000FC4 HHCCP008I CPU0000 thread ended: tid=0000049C,pid=232
2007/06/30 2007/06/30 14:49:57.044 00000FC4 HHCCP008I CPU0001 thread ended: tid=0000046C,pid=232
2007/06/30 2007/06/30 14:49:57.044 00000FC4 HHCCF047I Subchannel 0000 detached
2007/06/30 2007/06/30 14:49:57.044 00000FC4 HHCCF047I Subchannel 0001 detached
2007/06/30 2007/06/30 14:49:57.044 00000FC4 HHCCF047I Subchannel 0002 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCTT003I Timer thread ended
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0003 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0004 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0005 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0006 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0007 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0008 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0009 detached
```

```
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 000A detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 000B detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 000C detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 000D detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 000E detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 000F detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0010 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0011 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0012 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0013 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCTE004I Console connection thread terminated
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCDA023I 0300 cache hits 812, misses 114, waits 0
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0014 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCDA023I 0310 cache hits 5, misses 7, waits 0
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0015 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCDA023I 0320 cache hits 271, misses 35, waits 0
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0016 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCDA023I 0330 cache hits 137, misses 59, waits 0
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0017 detached
.
.
.
2007/06/30 2007/06/30 14:49:57.075 00000FC4 HHCCF047I Subchannel 0028 detached
2007/06/30 2007/06/30 14:49:57.075 00000FC4 HHCCF047I Subchannel 0029 detached
2007/06/30 2007/06/30 14:49:57.075 00000FC4 HHCCF047I Subchannel 002A detached
2007/06/30 2007/06/30 14:49:57.075 00000FC4 HHCCF047I Subchannel 002B detached
2007/06/30 2007/06/30 14:49:57.075 00000FC4 HHCCF047I Subchannel 002C detached
2007/06/30 2007/06/30 14:49:57.075 00000FC4 HHCCF047I Subchannel 002D detached
2007/06/30 2007/06/30 14:49:57.075 00000FC4 HHCCF047I Subchannel 002E detached
2007/06/30 2007/06/30 14:49:59.841 00000FC4 HHCCF047I Subchannel 002F detached
2007/06/30 2007/06/30 14:49:59.841 00000FC4 HHCIN902I Configuration release complete
2007/06/30 2007/06/30 14:49:59.841 00000FC4 HHCIN903I Calling termination routines
2007/06/30 2007/06/30 14:49:59.841 00000FC4 HHCHD900I Begin shutdown sequence
2007/06/30 2007/06/30 14:49:59.841 00000FC4 HHCHD901I Calling console_shutdown
2007/06/30 2007/06/30 14:49:59.841 00000FC4 HHCHD902I console_shutdown complete
2007/06/30 2007/06/30 14:49:59.841 00000FC4 HHCHD901I Calling hdl_term
2007/06/30 2007/06/30 14:49:59.841 00000FC4 HHCHD950I Begin HDL termination sequence
2007/06/30 2007/06/30 14:49:59.841 00000FC4 HHCHD951I Calling module dyngui cleanup routine
2007/06/30 2007/06/30 14:49:59.950 00000FC4 HHCHD952I Module dyngui cleanup complete
2007/06/30 2007/06/30 14:49:59.950 00000FC4 HHCHD951I Calling module *Hercules cleanup routine
2007/06/30 2007/06/30 14:49:59.950 00000FC4 HHCHD952I Module *Hercules cleanup complete
2007/06/30 2007/06/30 14:49:59.950 00000FC4 HHCHD959I HDL Termination sequence complete
2007/06/30 2007/06/30 14:49:59.950 00000FC4 HHCHD902I hdl_term complete
2007/06/30 2007/06/30 14:49:59.950 00000FC4 HHCHD901I Calling logger_term
2007/06/30 2007/06/30 14:49:59.950 00000FC4 HHCLG014I logger thread terminating
2007/06/30 2007/06/30 14:49:59.950 00000FC4 HHCLG014I logger thread terminating
2007/06/30 2007/06/30 14:49:59.950 00000FC4 HHCIN099I Hercules terminated
```

**Figure 66: EXIT command**

## 7.36 EXT (Generate external interrupt)

### *7.36.1 Function*

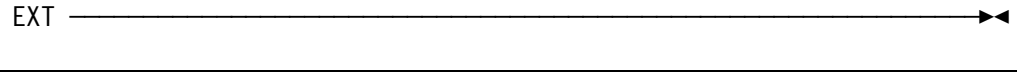
The EXT command generates an external interrupt (the virtual INTERRUPT key is pressed).

### *7.36.2 Syntax*

**Descriptive**

EXT

**Diagram**




### *7.36.3 Parameter*

None.

### *7.36.4 Examples*

**Example 1:**

Generate an external interrupt.

```
2007/06/30 15:30:01.015 00000FC4 ext
2007/06/30 15:30:01.015 00000FC4 HHCPN050I Interrupt key depressed
```

**Figure 67: EXT command**

## 7.37 F{+/-} (Mark frames usable or unusable)

### *7.37.1 Function*

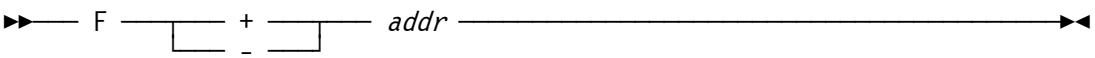
The F{+/-} command is used to mark storage frames as usable or unusable.

### *7.37.2 Syntax*

**Descriptive**

F{+ | -} *addr*

**Diagram**




### *7.37.3 Parameter*

**+** Mark storage frame as usable.

**-** Mark storage frame as unusable.

**addr** The address of the storage frame that has to be marked usable or unusable.

### *7.37.4 Examples*

**Example 1:**

Mark frame at address x'01000000' as unusable.

```
2007/06/30 14:13:14.265 0000073C f- 01000000
2007/06/30 14:13:14.265 0000073C HHCPN131I Frame 01000000 marked unusable
```

**Figure 68: F- command**

**Example 2:**

Mark frame at address x'01000000' as usable.

```
2007/06/30 14:13:25.390 0000073C f+ 01000000
2007/06/30 14:13:25.390 0000073C HHCPN131I Frame 01000000 marked usable
```

**Figure 69: F+ command**

## 7.38 FPC (Display floating point control register)

### *7.38.1 Function*

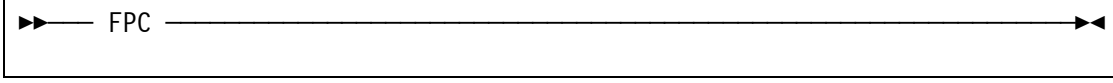
The FPC command displays the content of the floating point control register.

### *7.38.2 Syntax*

**Descriptive**

```
FPC
```

**Diagram**




### *7.38.3 Parameter*

None.

### *7.38.4 Examples*

**Example 1:**

Display the floating point control register.

```
2007/06/30 12:14:14.968 00000FC4 fpc
2007/06/30 12:14:14.968 00000FC4 FPC=00000000
```

**Figure 70: FPC command**

## 7.39 FPR (Display floating point registers)

### *7.39.1 Function*

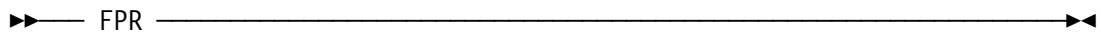
The FPR command displays the actual contents of the floating point registers (basic format). If the AFP bit is set in CR0 then additional floating point registers are displayed (extended format).

### *7.39.2 Syntax*

**Descriptive**

FPR

**Diagram**




### *7.39.3 Parameter*

None.

### *7.39.4 Examples*

**Example 1:**

Display the floating point registers.

```
2007/08/25 18:07:09.263 00000E18 fpr
2007/08/25 18:07:09.263 00000E18 CPU0000: FPR0=00000000 00000000  FPR2=00000000 00000000
2007/08/25 18:07:09.263 00000E18 CPU0000: FPR4=00000000 00000000  FPR6=00000000 00000000
```

**Figure 71: FPR command (basic format)**

**Example 2:**

Display the floating point registers when the AFP bit is set in CR0.

```
2007/08/25 18:07:19.263 00000E18 fpr
2007/08/25 18:07:19.263 00000E18 CPU0000: FPR0=00000000 00000000  FPR2=00000000 00000000
2007/08/25 18:07:19.263 00000E18 CPU0000: FPR1=00000000 00000000  FPR3=00000000 00000000
2007/08/25 18:07:19.263 00000E18 CPU0000: FPR4=00000000 00000000  FPR6=00000000 00000000
2007/08/25 18:07:19.263 00000E18 CPU0000: FPR5=00000000 00000000  FPR7=00000000 00000000
2007/08/25 18:07:19.263 00000E18 CPU0000: FPR8=00000000 00000000  FP10=00000000 00000000
2007/08/25 18:07:19.263 00000E18 CPU0000: FPR9=00000000 00000000  FP11=00000000 00000000
2007/08/25 18:07:19.263 00000E18 CPU0000: FP12=00000000 00000000  FP14=00000000 00000000
2007/08/25 18:07:19.263 00000E18 CPU0000: FP13=00000000 00000000  FP15=00000000 00000000
```

**Figure 72: FPR command (extended format)**

## 7.40 G (Turn off instruction stepping and start CPU)

### *7.40.1 Function*

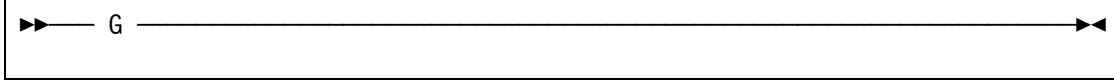
The G command turns off the instruction stepping and starts the CPU. This command has also to be used to continue normal processing after stopping the instruction stepping with the S- command.

### *7.40.2 Syntax*

**Descriptive**

G

**Diagram**




### *7.40.3 Parameter*

None.

### *7.40.4 Examples*

**Example 1:**

Stop the instruction stepping and start the CPU.

```
13:52:59.375 00000FC4 g
13:52:59.375 00000FC4 HHCCP014I CPU0000: Special-operation exception CODE=0013 ILC=4
13:52:59.375 00000FC4 CPU0000:  PSW=00080000 80000606 INST=B2790000     SACF  0(0)
13:52:59.375 00000FC4 CPU0000:  R:00000000:K:06=00080000 800005EC 40404040 F0F04040 ........  00
13:52:59.375 00000FC4 CPU0000: GR00=00000000  GR01=00000000  GR02=00000000  GR03=00000000
13:52:59.375 00000FC4 CPU0000: GR04=00000000  GR05=00000000  GR06=00000000  GR07=00000000
13:52:59.375 00000FC4 CPU0000: GR08=00000000  GR09=00000000  GR10=00000000  GR11=00000000
13:52:59.375 00000FC4 CPU0000: GR12=00000000  GR13=00000000  GR14=00000000  GR15=00000000
13:52:59.375 00000FC4 CPU0000: CR00=01B00200  CR01=00000000  CR02=00000000  CR03=00000000
13:52:59.375 00000FC4 CPU0000: CR04=00000000  CR05=00000000  CR06=FE000000  CR07=00000000
13:52:59.375 00000FC4 CPU0000: CR08=00000000  CR09=00000000  CR10=00000000  CR11=00000000
13:52:59.375 00000FC4 CPU0000: CR12=00000000  CR13=00000000  CR14=C2000000  CR15=00000000
13:52:59.390 00000FC4 HHCCD001I Readahead thread 1 started: tid=000000DC, pid=2068
13:52:59.390 00000FC4 HHCCD001I Readahead thread 2 started: tid=00000B6C, pid=2068
```

**Figure 73: G command**

## 7.41 GPR (Display or alter general purpose registers)

### *7.41.1 Function*

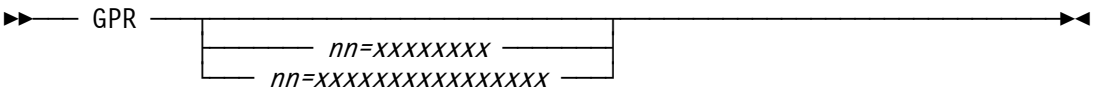
The GPR command displays or alters the actual content of the general purpose registers.

### *7.41.2 Syntax*

**Descriptive**

GPR [*nn=xxxxxxxx* | *nn=xxxxxxxxxxxxxxxx*]

**Diagram**




### *7.41.3 Parameter*

**nn** This specifies the optional register number (00-15) to be altered.

**xxxxxxxx** This is the register value in hexadecimal (1-8 hex digits for 32-bit registers, 1-16 hex digits for 64-bit registers).

### *7.41.4 Examples*

**Example 1:**

Display the general purpose registers.

```
16:54:31.625 00000FC4 gpr
16:54:31.625 00000FC4 CPU0000: GR00=00000000  GR01=00000000  GR02=0252E12C  GR03=02501EB0
16:54:31.625 00000FC4 CPU0000: GR04=02502400  GR05=00000C00  GR06=01379E76  GR07=00000000
16:54:31.625 00000FC4 CPU0000: GR08=0248EF58  GR09=00000000  GR10=0252EF7F  GR11=0252DF80
16:54:31.625 00000FC4 CPU0000: GR12=813799C8  GR13=0252DF80  GR14=81379D22  GR15=00000C00
```

**Figure 74: GPR command (display general purpose registers)**

**Example 2:**

Alter general purpose register number 15.

```
16:54:41.625 00000FC4 gpr 15=FF000000
16:54:41.625 00000FC4 CPU0000: GR00=00000000  GR01=00000000  GR02=0252E12C  GR03=02501EB0
16:54:41.625 00000FC4 CPU0000: GR04=02502400  GR05=00000C00  GR06=01379E76  GR07=00000000
16:54:41.625 00000FC4 CPU0000: GR08=0248EF58  GR09=00000000  GR10=0252EF7F  GR11=0252DF80
16:54:41.625 00000FC4 CPU0000: GR12=813799C8  GR13=0252DF80  GR14=81379D22  GR15=FF000000
```

**Figure 75: GPR command (alter general purpose register)**

## 7.42 HAO (Hercules Automatic Operator)

### *7.42.1 Function*

The Hercules Automatic operator (HAO) feature is a facility that allows to automatically issue console commands in response to certain messages being issued. To use the HAO facility it is necessary to define a rule, consisting of a target and an associated command.

The Hercules Automatic Operator is only for those messages issued by Hercules to its console. It can not be used for messages issued from the guest operating system.

The current implementation limits the total number of defined rules to 64. There is currently no way to define a command whose arguments varies based on actual message text. All of the defined rules are checked for a match each time Hercules issues a message, there is no way at this time to stop the processing of subsequent rules.

### *7.42.2 Syntax*

**Descriptive**

HAO TGT *target*

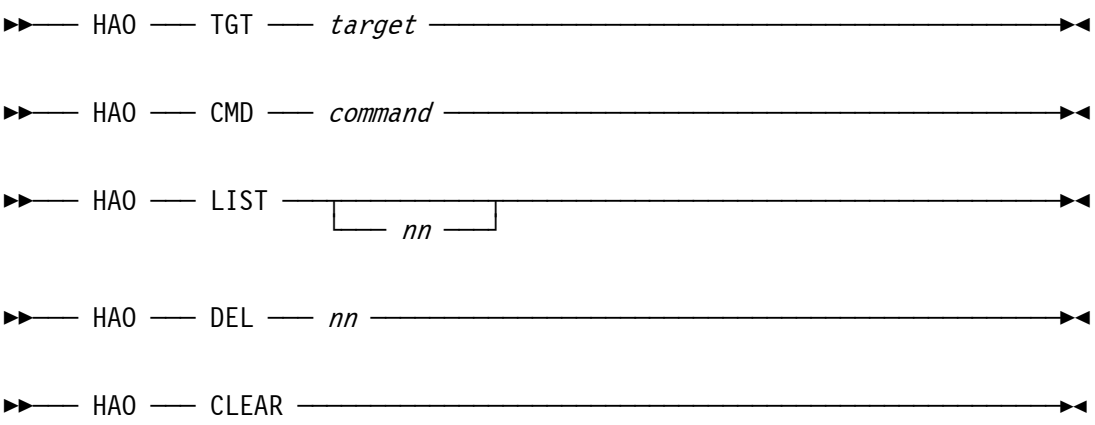
HAO CMD *command*

HAO LIST [*nn*]

HAO DEL *nn*

HAO CLEAR

**Diagram**



## 7.42.3 Parameter

**TGT** Keyword to define a new rule (pattern).

**target** Specifies the rule (pattern) to react on. The target is a regular expression pattern which is matched against the text of the messages that Hercules issues.

**CMD** Keyword to specify a command for a previously defined rule.

**command** Specifies the command to be executed if a target rule matches. The associated command must be a valid Hercules console command.

**LIST** List all rules/commands or list only rule/command at index *nn*.

**DEL** Delete rule and command at index *nn*.

**nn** Number of the index to be listed or deleted.

**CLEAR** Specifies to delete all defined rules. This stops the Automatic Operator.

## 7.42.4 Examples

**Example 1:**

Define target rule: Check for message HHCPN139E ("Command not found").

```
2007/06/20 16:54:31.625 00000FC4 hao tgt HHCPN139E
2007/06/20 16:54:31.625 00000FC4 HHCAO016I Target placed at index 0
```

**Figure 76: HAO command (tgt).**

**Example 2:**

Define command for target rule: Issue "?" command (List all valid commands).

```
2007/06/20 16:54:31.625 00000FC4 hao cmd ?
2007/06/20 16:54:31.625 00000FC4 HHCAO020I Command placed at index 0
```

**Figure 77: HAO command (cmd)**

**Example 3:**

List all defined rules.

```
2007/06/20 16:54:31.625 00000FC4 hao list
2007/06/20 16:54:31.625 00000FC4 HHCAO004I The defined Automatic Operator rule(s) are:
2007/06/20 16:54:31.625 00000FC4 HHCAO005I 00: 'HHCPN139E' -> '?'
2007/06/20 16:54:31.625 00000FC4 HHCAO006I 1 rule(s) displayed
```

**Figure 78: HAO command (list)**

**Example 4:**

Delete rule at index 0.

```
2007/06/20 16:54:31.625 00000FC4 hao del 0
2007/06/20 16:54:31.625 00000FC4 HHCAO025I Rule at index 0 succesfully deleted
```

**Figure 79: HAO command (del)**

**Example 5:**

Delete all rules (stop Hercules Automatic Operator).

```
2007/06/20 16:54:31.625 00000FC4 hao clear
2007/06/20 16:54:31.625 00000FC4 HHCAO022I All automatic operation rules cleared
```

**Figure 80: HAO command (clear)**

**Example 6:**

Sample of automatic execution of a command (triggerred via the use of an unknown console command, as defined in example 1 and 2).

```
2007/06/20 16:54:31.625 00000FC4 offload
2007/06/20 16:54:31.625 00000FC4 HHCPN139E Command "offload" not found; enter '?' for list.
2007/06/20 16:54:31.625 00000FC4 HHCAO003I Firing command '?'
2007/06/20 16:54:31.625 00000FC4 ?
2007/06/20 16:54:31.625 00000FC4 HHCPN140I Valid panel commands are…
2007/06/20 16:54:31.625 00000FC4
.
.
.
```

**Figure 81: HAO fired command**

## 7.43 HELP (List all commands / command specific help)

### *7.43.1 Function*

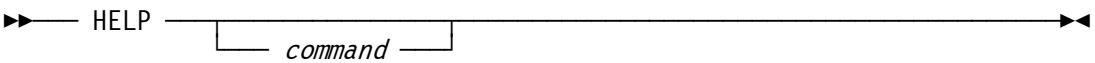
The HELP command displays additional information for specific Hercules console commands if extra information is available. If the command has additional help text it will be displayed. HELP without parameter lists all available commands (see also "?" command). The help text is usually limited to explaining the format of the command and its various required or optional parameters and is not meant to replace reading the documentation.

### *7.43.2 Syntax*

**Descriptive**

HELP [*command*]

**Diagram**




### *7.43.3 Parameter*

**command** The Hercules console command to which additional information is desired.

### *7.43.4 Examples*

**Example 1:**

Display help for a command with additional help information.

```
00000FC4 help ostailor
00000FC4 Format: "ostailor quiet | os/390 | z/os | vm | vse | linux | null". Specifies
00000FC4 the intended operating system. The effect is to reduce control panel message
00000FC4 traffic by selectively suppressing program check trace messages which are
00000FC4 considered normal in the specified environment. 'quiet' suppresses all
00000FC4 exception messages, whereas 'null' suppresses none of them. The other options
00000FC4 suppress some messages and not others depending on the specified o/s.
00000FC4 SEE ALSO the 'pgmtrace' command which allows you to further fine tune
00000FC4 the tracing of program interrupt exceptions.
```

**Figure 82: HELP command (information available)**

**Example 2:**

Display help for a command without additional help information.

```
2007/06/30 16:54:31.625 00000FC4 help quiet
2007/06/30 16:54:31.625 00000FC4 HHCPN142I No additional help available.
```

**Figure 83: HELP command (no information available)**

## 7.44 HERC (Send Hercules command)

### *7.44.1 Function*

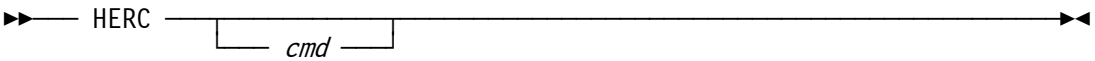
The HERC command sends a command in any CMDTGT mode to the Hercules Emulator. See also CMDTGT, SCP and PSCP commands.

### *7.44.2 Syntax*

**Descriptive**

HERC [*cmd*]

**Diagram**




### *7.44.3 Parameter*

**cmd** This is the command to be sent to the Hercules Emulator.

### *7.44.4 Examples*

**Example 1:**

Send a Hercules command in SCP or PSCP command target mode..

```
2008/08/30 09:10:04.390 00000FC4 herc psw
2008/08/30 09:10:04.390 00000FC4 psw sm=00 pk=0 cmwp=0 as=pri cc=0 pm=0 am=24 ia=0
2008/08/30 09:10:04.390 00000FC4 PSW=00000000 00000000
```

**Figure 84: HERC command**

## 7.45 HERCLOGO (Read new logo file)

### 7.45.1 Function

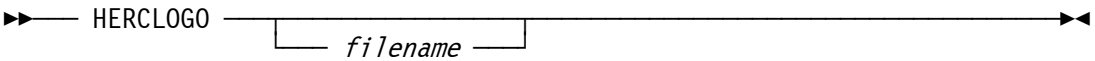
The HERCLOGO command loads a new logo file for 3270 terminal sessions. For details on how to code the logo file see chapter 10.

### 7.45.2 Syntax

**Descriptive**

HERCLOGO [*filename*]

**Diagram**




### 7.45.3 Parameter

**filename** The filename of a logo text file, located in the current Hercules directory, that will be loaded. If no filename is specified the built-in logo is used instead.

### 7.45.4 Examples

**Example 1:**

Load a new logo file.

```
2007/06/30 12:45:53.859 00000FC4 herclogo herclogo.txt
```

**Figure 85: HERCLOGO command**

## 7.46 HST (History of commands)

### *7.46.1 Function*

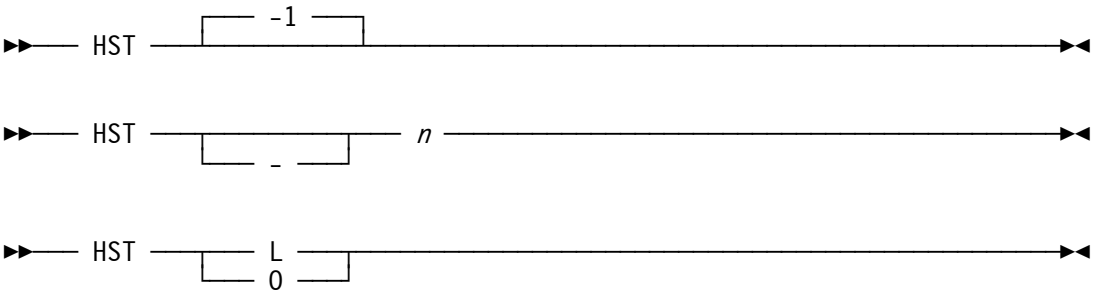
The HST command displays a list of the last commands entered from the command line.

### *7.46.2 Syntax*

**Descriptive**

```
HST [ -1 ]
HST [-]n
HST {L | 0}
```

**Diagram**




### *7.46.3 Parameter*

| | |
|---|---|
| **-1** | HST -1 is the same as HST without argument and retrieves the last entered command. |
| **[-]n** | If n is a positive number then HST retrieves the n-th command from the list. If n is a negative number then HST retrieves the n-th last command from the list. |
| **L or 0** | HST L or HST 0 (number "0") displays a list of the last ten commands entered on the command line. |

## *7.46.4 Examples*

**Example 1:**

Display a list of the last ten commands entered.

```
2007/06/30 14:23:48.809 00000FC4 hst 0
2007/06/30 14:23:48.809 00000FC4   11 ipl 0A80
2007/06/30 14:23:48.809 00000FC4   12 ?
2007/06/30 14:23:48.809 00000FC4   13 aea
2007/06/30 14:23:48.809 00000FC4   14 aia
2007/06/30 14:23:48.809 00000FC4   15 ar
2007/06/30 14:23:48.809 00000FC4   16 clocks
2007/06/30 14:23:48.809 00000FC4   17 cr
2007/06/30 14:23:48.809 00000FC4   18 fpr
2007/06/30 14:23:48.809 00000FC4   19 gpr
2007/06/30 14:23:48.809 00000FC4   20 help maxrates
```

**Figure 86: HST command**

## 7.47 I (Generate I/O attention interrupt for device)

### *7.47.1 Function*

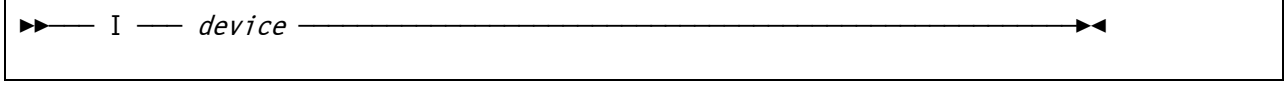
The I command generates an I/O attention interrupt for a certain device.

### *7.47.2 Syntax*

**Descriptive**

I *device*

**Diagram**




### *7.47.3 Parameter*

**device** The number of the device for which an I/O attention interrupt will be generated.

### *7.47.4 Examples*

**Example 1:**

Generate an I/O attention interrupt for device 0148.

```
2007/06/30 15:29:37.062 00000FC4 i 0148
2007/06/30 15:29:37.062 00000FC4 HHCPN045I Device 0148 attention request raised
```

**Figure 87: I command**

## 7.48 IODELAY (Display or set I/O delay value)

### *7.48.1 Function*

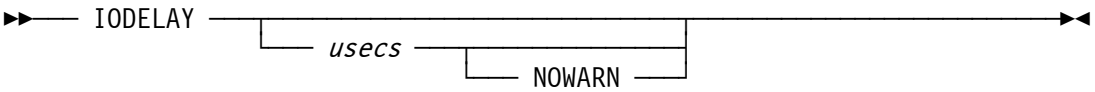
The IODELAY command displays the current setting or sets the amount of time in microseconds to wait after an I/O interrupt is ready to be set pending. The purpose of this parameter is to bypass a bug in the Linux/390 and z/Linux dasd.c device driver. The problem is more apt to happen under Hercules than on a real machine, as Hercules may present an I/O interrupt sooner than a real machine.

### *7.48.2 Syntax*

**Descriptive**

```
IODELAY [usecs [NOWARN]]
```

**Diagram**




### *7.48.3 Parameter*

**usecs** Amount of time in microseconds to wait after an I/O interrupt is ready to be set pending.

**NOWARN** If the IODELAY value is non-zero a warning message (HHCCF037W) will be issued unless NOWARN is specified.

### *7.48.4 Examples*

**Example 1:**

Display the current IODELAY value.

```
2007/06/30 17:14:44.406 00000FC4 iodelay
2007/06/30 17:14:44.406 00000FC4 HHCPN030I I/O delay = 0
```

**Figure 88: IODELAY command (display value)**

**Example 2:**

Change the current IODELAY value.

```
2007/06/30 17:15:26.218 00000FC4 iodelay 50
2007/06/30 17:15:26.218 00000FC4 HHCPN030I I/O delay = 50
```

**Figure 89: IODELAY command (set value)**

## 7.49 IPENDING (Display pending interrupts)

### *7.49.1 Function*

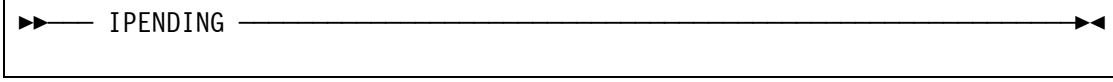
The IPENDING command displays the pending interrupts for each CPU and some lock information.

### *7.49.2 Syntax*

**Descriptive**

```
IPENDING
```

**Diagram**




### *7.49.3 Parameter*

None

### *7.49.4 Examples*

**Example 1:**

Display pending interrupts.

```
14:24:05.341 00000FC4 ipending
14:24:05.341 00000FC4 HHCPN123I CPU0000: CPUint=00000001 (State:00000001)&(Mask:DF00EE4F)
14:24:05.341 00000FC4           CPU0000: Interrupt not pending
14:24:05.341 00000FC4           CPU0000: I/O interrupt not pending
14:24:05.341 00000FC4           CPU0000: Clock comparator not pending
14:24:05.341 00000FC4           CPU0000: CPU timer not pending
14:24:05.341 00000FC4           CPU0000: Interval timer not pending
14:24:05.341 00000FC4           CPU0000: External call not pending
14:24:05.341 00000FC4           CPU0000: Emergency signal not pending
14:24:05.341 00000FC4           CPU0000: Machine check interrupt not pending
14:24:05.341 00000FC4           CPU0000: Service signal not pending
14:24:05.341 00000FC4           CPU0000: CPU interlock not held
14:24:05.341 00000FC4           CPU0000: lock not held
14:24:05.341 00000FC4           CPU0000: state STOPPED
14:24:05.341 00000FC4           CPU0000: instcount 2139645750
14:24:05.341 00000FC4           CPU0000: siocount 52968
14:24:05.341 00000FC4           CPU0000: psw 070e0000 00000000
14:24:05.341 00000FC4 HHCPN123I CPU0001: CPUint=00000001 (State:00000001)&(Mask:DF00EE4F)
```

```
14:24:05.341 00000FC4           CPU0001: Interrupt not pending
14:24:05.341 00000FC4           CPU0001: I/O interrupt not pending
14:24:05.341 00000FC4           CPU0001: Clock comparator not pending
14:24:05.341 00000FC4           CPU0001: CPU timer not pending
14:24:05.341 00000FC4           CPU0001: Interval timer not pending
14:24:05.341 00000FC4           CPU0001: External call not pending
14:24:05.341 00000FC4           CPU0001: Emergency signal not pending
14:24:05.341 00000FC4           CPU0001: Machine check interrupt not pending
14:24:05.341 00000FC4           CPU0001: Service signal not pending
14:24:05.341 00000FC4           CPU0001: CPU interlock not held
14:24:05.341 00000FC4           CPU0001: lock not held
14:24:05.341 00000FC4           CPU0001: state STOPPED
14:24:05.341 00000FC4           CPU0001: instcount 1263569267
14:24:05.341 00000FC4           CPU0001: siocount 20795
14:24:05.341 00000FC4           CPU0001: psw 070e0000 00000000
14:24:05.341 00000FC4 HHCPN123I CPU0002: offline
14:24:05.341 00000FC4 HHCPN123I CPU0003: offline
14:24:05.341 00000FC4 HHCPN123I CPU0004: offline
14:24:05.341 00000FC4 HHCPN123I CPU0005: offline
14:24:05.341 00000FC4 HHCPN123I CPU0006: offline
14:24:05.341 00000FC4 HHCPN123I CPU0007: offline
14:24:05.341 00000FC4           Config mask 00000003 started mask 00000003 waiting mask 00000003
14:24:05.341 00000FC4           Broadcast count 0 code 4
14:24:05.341 00000FC4           Signaling facility not busy
14:24:05.341 00000FC4           TOD lock not held
14:24:05.341 00000FC4           Main lock not held
14:24:05.341 00000FC4           Int lock not held
14:24:05.341 00000FC4           DEV0AE0: suspended
14:24:05.341 00000FC4           DEV0E21: busy local
14:24:05.341 00000FC4           I/O interrupt queue: (NULL)
```

**Figure 90: IPENDING command**

## 7.50 IPL (IPL normal from device xxxx)

### *7.50.1 Function*

This console command performs the Initial Program Load (IPL) manual function from the given device. The IPL command may also be used to perform a load from CD-ROM or server. For example if a standard SUSE S/390 Linux distribution CD is loaded and mounted on /cdrom, this CD-ROM may then be IPL'ed using *"IPL /cdrom/suse.ins"*.

If the first operand is a device name (devnum is a 1- to 4-digit hexadecimal number), a CCW-type IPL is initiated from the indicated device number and SCLP disk I/O is disabled.

If the first operand is a filename a list-directed IPL is performed from the specified ".ins" file and SCLP disk I/O is enabled for the directory path where the ".ins" file is located.

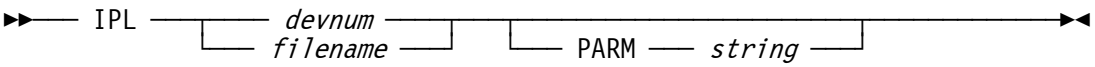
An optional 'PARM' keyword followed by a string can also be passed to the IPL command processor. The string will be loaded into the low-order 32 bits of the general purpose registers (4 characters per register for up to 64 bytes). The PARM option behaves similarly to the VM IPL command.

### *7.50.2 Syntax*

**Descriptive**

IPL {*devnum* | *filename*} [PARM *string*]

**Diagram**




### *7.50.3 Parameter*

**devnum** The device address from which the emulator will IPL the system.

**filename** The name of a ".ins" file to be loaded.

**PARM** Optional PARM keyword.

**string** The character string (max. 64 bytes) to be passed to the IPL command processor. The character string will be loaded into the low-order 32 bits of the GPRs (4 characters per register).

## *7.50.4 Examples*

**Example 1:**

IPL normal from device 0148.

```
ipl 0148
HHCCP048I 0011:CCW=03000000 20000001=>040C0000 0001CEFA 00000000 00000000 ................
HHCCP075I 0011:Stat=0200 Count=0001
HHCCP076I 0011:Sense=40000000 00000000 00000000 00000000 00000000 00000000
HHCCP077I 0011:Sense=INTREQ
HHCCP048I 00C1:CCW=03000000 20000001=>040C0000 0001CEFA 00000000 00000000 ................
HHCCP075I 00C1:Stat=0200 Count=0001
HHCCP076I 00C1:Sense=40000000 00000000 00000000 00000000 00000000 00000000
HHCCP077I 00C1:Sense=INTREQ
HHCCP048I 00C2:CCW=03000000 20000001=>040C0000 0001CEFA 00000000 00000000 ................
HHCCP075I 00C2:Stat=0200 Count=0001
HHCCP076I 00C2:Sense=40000000 00000000 00000000 00000000 00000000 00000000
HHCCP077I 00C2:Sense=INTREQ
HHCCP075I 00C3:Stat=0200 Count=0001
HHCCP076I 00C3:Sense=40000000 00000000 00000000 00000000 00000000 00000000
HHCCP077I 00C3:Sense=INTREQ
HHCCP048I 00C4:CCW=03000000 20000001=>040C0000 0001CEFA 00000000 00000000 ................
HHCCP075I 00C4:Stat=0200 Count=0001
HHCCP076I 00C4:Sense=40000000 00000000 00000000 00000000 00000000 00000000
HHCCP077I 00C4:Sense=INTREQ
HHCCP048I 00C5:CCW=03000000 20000001=>040C0000 0001CEFA 00000000 00000000 ................
HHCCP075I 00C5:Stat=0200 Count=0001
HHCCP076I 00C5:Sense=40000000 00000000 00000000 00000000 00000000 00000000
HHCCP077I 00C5:Sense=INTREQ
HHCCP048I 00C6:CCW=03000000 20000001=>040C0000 0001CEFA 00000000 00000000 ................
HHCCP075I 00C6:Stat=0200 Count=0001
HHCCP076I 00C6:Sense=40000000 00000000 00000000 00000000 00000000 00000000
HHCCP077I 00C6:Sense=INTREQ
HHCCP048I 00C7:CCW=03000000 20000001=>040C0000 0001CEFA 00000000 00000000 ................
HHCCP075I 00C7:Stat=0200 Count=0001
HHCCP076I 00C7:Sense=40000000 00000000 00000000 00000000 00000000 00000000
HHCCP077I 00C7:Sense=INTREQ
HHCCP048I 0480:CCW=03000000 20000001=>040C0000 0001CEFA 00000000 00000000 ................
HHCCP075I 0480:Stat=0200 Count=0001
HHCCP076I 0480:Sense=40220000 00C00300 00000000 00800100 010000FF FF000000
HHCCP077I 0480:Sense=INTREQ EOC WRI
.
.
.
```

**Figure 91: IPL command**

## 7.51 IPLC (IPL clear from device xxxx)

### *7.51.1 Function*

This console command performs the Initial Program Load (IPL) clear manual function from the given device. The IPL command may also be used to perform a load from CD-ROM or server. For example if a standard SUSE S/390 Linux distribution CD is loaded and mounted on /cdrom, this CD-ROM may then be IPL'ed using *"IPL /cdrom/suse.ins"*.

If the first operand is a device name (devnum is a 1- to 4-digit hexadecimal number), a CCW-type IPL is initiated from the indicated device number and SCLP disk I/O is disabled.

If the first operand is a filename a list-directed IPL is performed from the specified ".ins" file and SCLP disk I/O is enabled for the directory path where the ".ins" file is located.

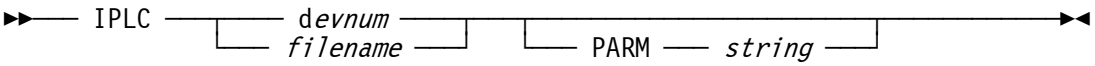
An optional 'PARM' keyword followed by a string can also be passed to the IPL command processor. The string will be loaded into the low-order 32 bits of the general purpose registers (4 characters per register for up to 64 bytes). The PARM option behaves similarly to the VM IPL command.

### *7.51.2 Syntax*

**Descriptive**

IPLC {*devnum* | *filename*} [PARM *string*]

**Diagram**




### *7.51.3 Parameter*

**devnum** The device address from which the emulator will IPL the system.

**filename** The name of a ".ins" file to be loaded.

**PARM** Optional PARM keyword.

**string** The character string (max. 64 bytes) to be passed to the IPL command processor. The character string will be loaded into the low-order 32 bits of the GPRs (4 characters per register).

## *7.51.4 Examples*

**Example 1:**

IPL clear from device 0148.

```
iplc 0148
HHCCP048I 0011:CCW=03000000 20000001=>040C0000 0001CEFA 00000000 00000000 ................
HHCCP075I 0011:Stat=0200 Count=0001
HHCCP076I 0011:Sense=40000000 00000000 00000000 00000000 00000000 00000000
HHCCP077I 0011:Sense=INTREQ
HHCCP048I 00C1:CCW=03000000 20000001=>040C0000 0001CEFA 00000000 00000000 ................
HHCCP075I 00C1:Stat=0200 Count=0001
HHCCP076I 00C1:Sense=40000000 00000000 00000000 00000000 00000000 00000000
HHCCP077I 00C1:Sense=INTREQ
HHCCP048I 00C2:CCW=03000000 20000001=>040C0000 0001CEFA 00000000 00000000 ................
HHCCP075I 00C2:Stat=0200 Count=0001
HHCCP076I 00C2:Sense=40000000 00000000 00000000 00000000 00000000 00000000
HHCCP077I 00C2:Sense=INTREQ
HHCCP075I 00C3:Stat=0200 Count=0001
HHCCP076I 00C3:Sense=40000000 00000000 00000000 00000000 00000000 00000000
HHCCP077I 00C3:Sense=INTREQ
HHCCP048I 00C4:CCW=03000000 20000001=>040C0000 0001CEFA 00000000 00000000 ................
HHCCP075I 00C4:Stat=0200 Count=0001
HHCCP076I 00C4:Sense=40000000 00000000 00000000 00000000 00000000 00000000
HHCCP077I 00C4:Sense=INTREQ
HHCCP048I 00C5:CCW=03000000 20000001=>040C0000 0001CEFA 00000000 00000000 ................
HHCCP075I 00C5:Stat=0200 Count=0001
HHCCP076I 00C5:Sense=40000000 00000000 00000000 00000000 00000000 00000000
HHCCP077I 00C5:Sense=INTREQ
HHCCP048I 00C6:CCW=03000000 20000001=>040C0000 0001CEFA 00000000 00000000 ................
HHCCP075I 00C6:Stat=0200 Count=0001
HHCCP076I 00C6:Sense=40000000 00000000 00000000 00000000 00000000 00000000
HHCCP077I 00C6:Sense=INTREQ
HHCCP048I 00C7:CCW=03000000 20000001=>040C0000 0001CEFA 00000000 00000000 ................
HHCCP075I 00C7:Stat=0200 Count=0001
HHCCP076I 00C7:Sense=40000000 00000000 00000000 00000000 00000000 00000000
HHCCP077I 00C7:Sense=INTREQ
HHCCP048I 0480:CCW=03000000 20000001=>040C0000 0001CEFA 00000000 00000000 ................
HHCCP075I 0480:Stat=0200 Count=0001
HHCCP076I 0480:Sense=40220000 00C00300 00000000 00800100 010000FF FF000000
HHCCP077I 0480:Sense=INTREQ EOC WRI
.
.
.
```

**Figure 92: IPLC command**

## 7.52 K (Display CCKD internal trace)

### *7.52.1 Function*

The K command displays the internal CCKD trace entries. Because the default value for the CCKD trace entries is set to zero, the trace must first be enabled by setting a trace entry value. This can be done either with the system parameter or the console command "CCKD TRACE=n", where n specifies the number of available trace entries and can be in the range of 1 to 200000.

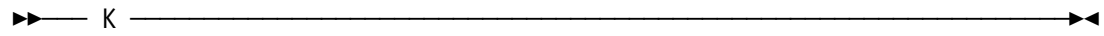
Please note that the K command can produce a lot of messages depending on the available trace entries.

### *7.52.2 Syntax*

**Descriptive**

K

**Diagram**




### *7.52.3 Parameter*

None.

### *7.52.4 Examples*

**Example 1:**

Display the current CCKD trace entries.

```
0000008E k
0000008E HHCCD900I print_itrace
0000008E 14:31:27.109347 0148:gcperc size 262144 1st 0xfb72c45 nbr 69 largest 16099545
0000008E 14:31:27.109371 0148:gcperc free[   0]:0fb72c45 end 0fb72cbe len        121*pend 0
0000008E 14:31:27.109382 0148:gcperc free[  67]:0fb72cbe end 0fb7ea56 len      48536 pend 1
0000008E 14:31:27.109392 0148:gcperc free[   1]:0fba543c end 0fba5f64 len       2856 pend 0
0000008E 14:31:27.109403 0148:gcperc free[   2]:0fba6094 end 0fba60bb len         39 pend 0
0000008E 14:31:27.109413 0148:gcperc free[   3]:0fbb7989 end 0fbb7a1b len        146 pend 0
0000008E 14:31:27.109421 0148:gcperc free[   4]:0fbb7ab4 end 0fbb7b26 len        114 pend 0
0000008E 14:31:27.109428 0148:gcperc free[   5]:0fbc971a end 0fbc975b len         65 pend 0
.
several lines not displayed
.
```

```
0000008E 14:31:27.109844 0148:gcperc selected space 0xfb7ea56 len 65778
0000008E 14:31:27.109850 0148:file[0] fd[5] read, off 0xfb7ea56 len 65778
0000008E 14:31:27.109942 0148:file[0] l2[99,111] trk[25455] read_l2ent 0x31f10
0000008E 14:31:27.109950 0148:file[0] read_l2 99 active 0 0 0
0000008E 14:31:27.109956 0148:l2[0,99] cache[82] hit
0000008E 14:31:27.109961 0148:file[0] l2[99,111] trk[25455] read_l2ent 0xfb7ea56 22094 22094
0000008E 14:31:27.109968 0148:gcperc move trk 25455 at pos 0xfb7ea56 len 22094
0000008E 14:31:27.109973 0148:file[0] trk[25455] write_trkimg len 22094 buf 08B0FF28:0106a10000
0000008E 14:31:27.109981 0148:file[0] read_l2 99 active 0 99 82
0000008E 14:31:27.109986 0148:file[0] trk[25455] write_trkimg oldl2 0xfb7ea56 22094 22094
0000008E 14:31:27.110006 0148:get_space len 22094 largest 16099545 flags 0x01
0000008E 14:31:27.110012 0148:get_space found 0x10f9c970 len 22094 size 22094
0000008E 14:31:27.110018 0148:file[0] fd[5] write, off 0x10f9c970 len 22094
0000008E 14:31:27.120709 0148:file[0] l2[99,111] trk[25455] write_l2ent 0x10f9c970 22094 22094
0000008E 14:31:27.120719 0148:file[0] fd[5] write, off 0x32288 len 8
0000008E 14:31:27.120734 0148:rel_space offset fb7ea56 len 22094 size 22094
0000008E 14:31:27.120743 0148:file[0] l2[127,0] trk[32512] read_l2ent 0x3ff10
0000008E 14:31:27.120749 0148:file[0] read_l2 127 active 0 99 82
0000008E 14:31:27.120755 0148:l2[0,127] cache[110] hit
0000008E 14:31:27.120759 0148:file[0] l2[127,0] trk[32512] read_l2ent 0xfb840a4 372 372
0000008E 14:31:27.120766 0148:gcperc move trk 32512 at pos 0xfb840a4 len 372
0000008E 14:31:27.120771 0148:file[0] trk[32512] write_trkimg len 372 buf 08B15576:0108770007
0000008E 14:31:27.120778 0148:file[0] read_l2 127 active 0 127 110
0000008E 14:31:27.120783 0148:file[0] trk[32512] write_trkimg oldl2 0xfb840a4 372 372
0000008E 14:31:27.120789 0148:get_space len 372 largest 16077451 flags 0x01
0000008E 14:31:27.120794 0148:get_space found 0xfba543c len 372 size 372
0000008E 14:31:27.120799 0148:file[0] fd[5] write, off 0xfba543c len 372
0000008E 14:31:27.120814 0148:file[0] l2[127,0] trk[32512] write_l2ent 0xfba543c 372 372
0000008E 14:31:27.120822 0148:file[0] fd[5] write, off 0x3ff10 len 8
0000008E 14:31:27.120837 0148:rel_space offset fb840a4 len 372 size 372
0000008E 14:31:27.120845 0148:file[0] l2[126,232] trk[32488] read_l2ent 0x3f710
0000008E 14:31:27.120852 0148:file[0] read_l2 126 active 0 127 110
0000008E 14:31:27.120857 0148:l2[0,126] cache[109] hit
0000008E 14:31:27.120862 0148:file[0] l2[126,232] trk[32488] read_l2ent 0xfb84218 373 373
0000008E 14:31:27.120869 0148:gcperc move trk 32488 at pos 0xfb84218 len 373
0000008E 14:31:27.120874 0148:file[0] trk[32488] write_trkimg len 373 buf 08B156EA:010875000d
0000008E 14:31:27.120881 0148:file[0] read_l2 126 active 0 126 109
0000008E 14:31:27.120886 0148:file[0] trk[32488] write_trkimg oldl2 0xfb84218 373 373
0000008E 14:31:27.120892 0148:get_space len 373 largest 16077451 flags 0x01
0000008E 14:31:27.120897 0148:get_space found 0xfba55b0 len 373 size 373
0000008E 14:31:27.120902 0148:file[0] fd[5] write, off 0xfba55b0 len 373
0000008E 14:31:27.120914 0148:file[0] l2[126,232] trk[32488] write_l2ent 0xfba55b0 373 373
0000008E 14:31:27.120922 0148:file[0] fd[5] write, off 0x3fe50 len 8
0000008E 14:31:27.120934 0148:rel_space offset fb84218 len 373 size 373
.
.
.
```

**Figure 93: K command**

## 7.53 LDMOD (Load a module)

### *7.53.1 Function*

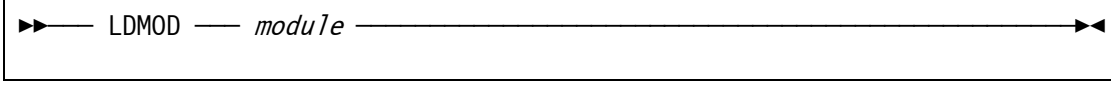
The LDMOD command loads a module.

### *7.53.2 Syntax*

**Descriptive**

LDMOD *module*

**Diagram**




### *7.53.3 Parameter*

**module** The name of the module that has to be loaded.

### *7.53.4 Examples*

**Example 1:**

Load module “dyncrypt”.

```
10:33:50.078 0000008E ldmod dyncrypt
10:33:50.078 0000008E HHCHD100I Loading dyncrypt ...
10:33:50.093 0000008E Crypto module loaded (c) Copyright Bernard van der Helm, 2003-2005
10:33:50.093 0000008E HHCHD101I Module dyncrypt loaded
```

**Figure 94: LDMOD command**

## 7.54 LOADCORE (Load a core image file)

### 7.54.1 Function

The LOADCORE command allows you to load a binary data file into real storage. This function is used mainly for emulator debugging purposes. A certain core snapshot that was saved can be restored at any later time to reproduce some tests with identical real storage values. The file *filename* is presumed to be a pure binary image file previously created via the SAVECORE command.

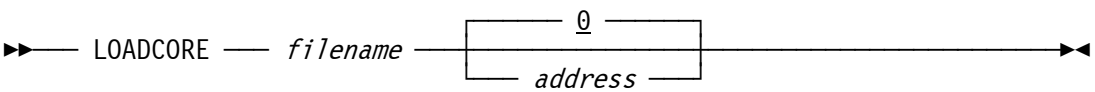
Please note that you must stop all CPUs (see STOP / STOPALL commands) before loading a core image from a file.

### 7.54.2 Syntax

**Descriptive**

LOADCORE *filename* [*address* | 0]

**Diagram**




### 7.54.3 Parameter

**filename** This argument specifies the file name (and optionally the path) of the file from which the core image will be loaded.

**address** Address specifies the start address of the real storage to where the saved file has to be loaded to. A value of zero, the default, means the beginning of real storage.

### 7.54.4 Examples

**Example 1:**

Load a core image file to address zero of real storage.

```
2007/06/30 09:52:20.203 0000008E stopall
2007/06/30 09:52:25.781 0000008E loadcore d:\core02.bin 0
2007/06/30 09:52:25.812 0000008E HHCPN112I Loading d:\core02.bin to location 0
2007/06/30 09:52:34.171 0000008E HHCPN113I 268431360 bytes read from d:\core02.bin
```

**Figure 95: LOADCORE command**

## 7.55 LOADPARM (Set IPL parameter)

### *7.55.1 Function*

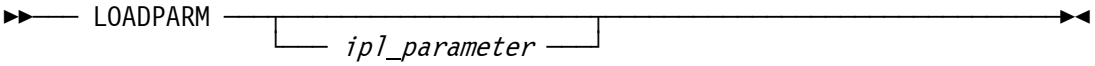
The LOADPARM command displays or changes the eight-character IPL parameter which is used by all MVS based operating systems (MVS 3.8J, MVS/SP, MVS/XA, MVS/ESA, OS/390, z/OS, …) to select the system parameter.

### *7.55.2 Syntax*

**Descriptive**

LOADPARM [*ipl_parameter*]

**Diagram**




### *7.55.3 Parameter*

**ipl_parameter** The system parameter used for the IPL of the intended operating system. The parameters are operating system dependent, the manuals of the operating system must be consulted for details.

### *7.55.4 Examples*

**Example 1:**

Display the current LOADPARM setting.

```
2007/06/30 08:15:03.875 00000FC4 loadparm
2007/06/30 08:15:03.875 00000FC4 HHCPN051I LOADPARM=014800M1
```

**Figure 96: LOADPARM command (display IPL parameter)**

**Example 2:**

Change the current LOADPARM setting.

```
2007/06/30 08:15:19.109 00000FC4 loadparm 014800t1
2007/06/30 08:15:19.109 00000FC4 HHCPN051I LOADPARM=014800T1
```

**Figure 97: LOADPARM command (set IPL parameter)**

## 7.56 LOADTEXT (Load a text deck file)

### *7.56.1 Function*

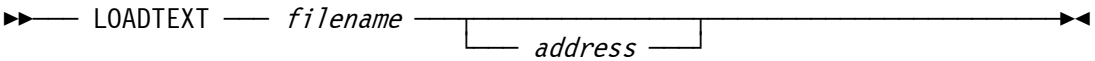
The LOADTEXT command is essentially identical to the LOADCORE command (see section 7.54 for details) except that it loads a text deck file with “TXT” and “END” 80 byte records (i.e. an object deck).

### *7.56.2 Syntax*

**Descriptive**

LOADTEXT *filename* [*address*]

**Diagram**




### *7.56.3 Parameter*

**filename** This argument specifies the file name (and optionally the path) of the file from which the text deck will be loaded.

**address** Address specifies the start address of the real storage to where the saved text deck file will be loaded.

### *7.56.4 Examples*

See LOADCORE command for similar examples.

## 7.57 LOG (Direct log output)

### *7.57.1 Function*

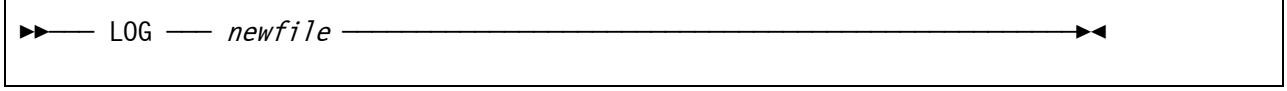
The LOG command lets you redirect the log output to another destination.

### *7.57.2 Syntax*

**Descriptive**

LOG *newfile*

**Diagram**




### *7.57.3 Parameter*

**newfile** The file name and optionally the path to which the log output has to be written to.

### *7.57.4 Examples*

**Example 1:**

Redirect the log output to a new log file destination.

```
2007/06/30 15:53:03.343 0000008E log d:/mvs/log/newlog.txt
2007/06/30 15:53:03.343 0000008E HHCLG018I log switched to d:/mvs/log/newlog.txt
```

**Figure 98: LOG command**

## 7.58 LOGOPT (Change log options)

### *7.58.1 Function*

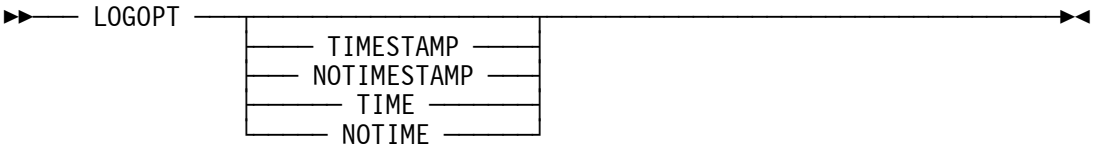
The LOGOPT command lets you change the logging options. "TIMESTAMP" inserts a time stamp in front of each log message, whereas "NOTIMESTAMP" displays log messages with no time stamps. Entering the command without any argument displays the current logging options.

### *7.58.2 Syntax*

**Descriptive**

```
LOGOPT [TIMESTAMP | NOTIMESTAMP | TIME | NOTIME]
```

**Diagram**




### *7.58.3 Parameter*

| | |
|---|---|
| **TIMESTAMP** | Insert a time stamp in front of each log message. This is the same as using TIME. |
| **NOTIMESTAMP** | Display log messages without a timestamp. This is the same as using NOTIME. |
| **TIME** | Insert a time stamp in front of each log message. This is the same as using TIMESTAMP. |
| **NOTIME** | Display log messages without a timestamp. This is the same as using NOTIMESTAMP. |

### *7.58.4 Examples*

**Example 1:**

Insert a time stamp in front of each log message.

```
2007/06/30 17:34:47.843 0000008E logopt timestamp
2007/06/30 17:34:47.843 0000008E HHCPN197I Log option set: TIMESTAMP
```

**Figure 99: LOGOPT command**

## 7.59 LPARNAME (Display or set LPAR name)

### 7.59.1 Function

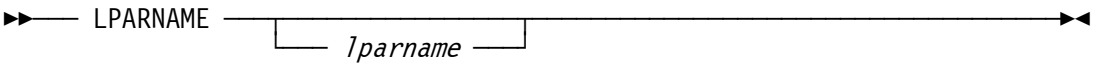
The LPARNAME command is used to display the current LPAR name or to define the LPAR name returned by the DIAG x'204'.

### 7.59.2 Syntax

**Descriptive**

LPARNAME [*lparname*]

**Diagram**




### 7.59.3 Parameter

**lparname** This specifies the new LPAR name. The value must be maximum 8 alphanumeric characters. If this parameter is omitted the current LPAR name is displayed.

### 7.59.4 Examples

**Example 1:**

Display the current LPAR name.

```
2009/01/15 05:58:16.781 00000FE4 lparname
2009/01/15 05:58:16.781 00000FE4 HHCPN056I LPAR name = HERCULES
```

**Figure 100: LPARNAME command (display LPAR name)**

**Example 2:**

Define a new LPAR name.

```
2009/01/15 05:59:24.212 00000FE4 lparname TESTLPAR
2009/01/15 05:59:24.212 00000FE4 lparname
2009/01/15 05:59:24.212 00000FE4 HHCPN056I LPAR name = TESTLPAR
```

**Figure 101: LPARNAME command (define LPAR name)**

## 7.60 LPARNUM (Display or set LPAR identification number)

### *7.60.1 Function*

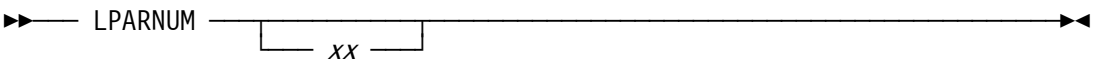
The LPARNUM console command displays or set the LPAR identification number. The LPAR identification number can be one or two hexadecimal digits. If one digit is specified then STIDP stores a format-0 CPU ID containing the logical CPU number, the LPAR number, and the last four digits of the CPU serial number. If two digits are specified then STIDP stores a format-1 CPU ID containing the two-digit LPAR identifier followed by the last four digits of the CPU serial number..

### *7.60.2 Syntax*

**Descriptive**

LPARNUM [*xx*]

**Diagram**




### *7.60.3 Parameter*

**xx** Specifies the one- or two-digit hexadecimal LPAR identification number stored by the STIDP instruction. If a one-digit number is specified then STIDP stores a format-0 CPU ID. If a two-digit number is specified then STIDP stores a format-1 CPU-ID..

### *7.60.4 Examples*

**Example 1:**

Set LPARNUM to 21.

```
2009/03/31 23:34:30.402 00000708 lparnum 21
```

**Figure 102: LPARNUM command (set LPARNUM).**

**Example 2:**

Display LPARNUM.

```
2009/03/31 23:34:33.402 00000708 lparnum
2009/03/31 23:34:33.402 00000708 HHCPN060I LPAR number = 21
```

**Figure 103: LPARNUM command (display LPARNUM).**

## 7.61 LSDEP (List module dependencies)

### *7.61.1 Function*

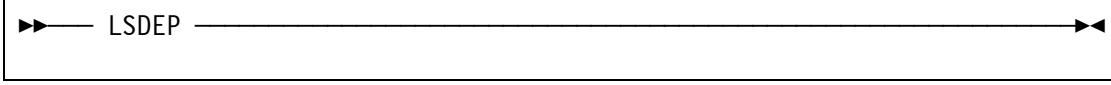
This function lists the Hercules module dependencies.

### *7.61.2 Syntax*

**Descriptive**

```
LSDEP
```

**Diagram**




### *7.61.3 Parameter*

None.

### *7.61.4 Examples*

**Example 1:**

List the module dependencies.

```
2007/06/30 14:24:39.637 00000FC4 lsdep
2007/06/30 14:24:39.637 00000FC4 dependency(HERCULES) version(3.03) size(5)
2007/06/30 14:24:39.637 00000FC4 dependency(REGS) version(3.03) size(38376)
2007/06/30 14:24:39.637 00000FC4 dependency(DEVBLK) version(3.03) size(1664)
2007/06/30 14:24:39.637 00000FC4 dependency(SYSBLK) version(3.03) size(39440)
2007/06/30 14:24:39.637 00000FC4 dependency(WEBBLK) version(2.17) size(24)
```

**Figure 104: LSDEP command**

## 7.62 LSMOD (List dynamic modules)

### *7.62.1 Function*

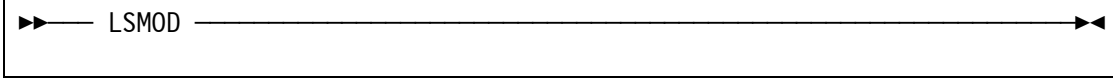
The LSMOD command lists all the loaded Hercules dynamic modules and shows details for each module.

### *7.62.2 Syntax*

**Descriptive**

```
LSMOD
```

**Diagram**




### *7.62.3 Parameter*

None.

### *7.62.4 Examples*

**Example 1:**

List the dynamic modules.

```
00000FC4 lsmod
00000FC4 dll type = load, name = hdt3088
00000FC4  symbol = debug_tt32_stats, loadcount = 1, owner = hdt3088
00000FC4  devtype = CTCI-WIN VMNET CTCT CTCI 3088 LCS
00000FC4 dll type = load, name = hdt3420
00000FC4  devtype = 3430 3422 8809 9348 9347 3490 3480 3420 3411 3410
00000FC4 dll type = load, name = hdt3270
00000FC4  devtype = 3287 3270 3215 1052
00000FC4 dll type = load, name = hdt1403
00000FC4  devtype = 3211 1403
00000FC4 dll type = load, name = hdt3525
00000FC4  devtype = 3525
00000FC4 dll type = load, name = hdt3505
00000FC4  devtype = 3505 2501 1442
00000FC4 dll type = load, name = dyngui
00000FC4  symbol = panel_command, loadcount = 2, owner = dyngui
00000FC4  symbol = debug_cpu_state, loadcount = 1, owner = dyngui
00000FC4  symbol = daemon_task, loadcount = 1, owner = dyngui
```

```
00000FC4  symbol = panel_display, loadcount = 1, owner = dyngui
00000FC4 dll type = load, name = dyncrypt
00000FC4  symbol = z900_compute_message_authentication_code, loadcount = 1, owner = dyncrypt
00000FC4  symbol = z900_compute_last_message_digest, loadcount = 1, owner = dyncrypt
00000FC4  symbol = z900_compute_intermediate_message_digest, loadcount = 1, owner = dyncrypt
00000FC4  symbol = z900_cipher_message_with_chaining, loadcount = 1, owner = dyncrypt
00000FC4  symbol = z900_cipher_message, loadcount = 1, owner = dyncrypt
00000FC4  symbol = s390_compute_message_authentication_code, loadcount = 1, owner = dyncrypt
00000FC4  symbol = s390_compute_last_message_digest, loadcount = 1, owner = dyncrypt
00000FC4  symbol = s390_compute_intermediate_message_digest, loadcount = 1, owner = dyncrypt
00000FC4  symbol = s390_cipher_message_with_chaining, loadcount = 1, owner = dyncrypt
00000FC4  symbol = s390_cipher_message, loadcount = 1, owner = dyncrypt
00000FC4 dll type = load, name = hdteq
00000FC4  symbol = hdl_device_type_equates, loadcount = 1, owner = hdteq
00000FC4 dll type = main, name = *Hercules, flags = (nounload)
00000FC4  symbol = panel_display, loadcount = 0, owner = *Hercules
00000FC4  symbol = panel_command, loadcount = 0, owner = *Hercules
00000FC4  symbol = parse_args, loadcount = 0, owner = *Hercules
00000FC4  devtype = 9336 9335 9332 9313 3370 3310 0671 9345 3390 3380 3375 3350 3340 3330 2314
                    2311
```

**Figure 105: LSMOD command**

## 7.63 MAXRATES (Display maximum observed MIPS/SIO rate or define new reporting interval)

### *7.63.1 Function*

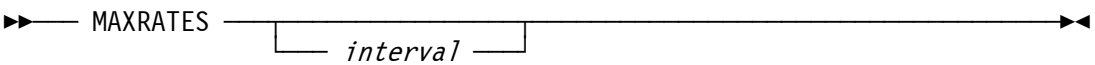
The MAXRATES command shows the maximum observed MIPS (million instructions per second) and SIO (Start I/O) rates reached during the current interval. It lets you also set the MAXRATES interval to another value than the default of 1440 minutes (1 day).

### *7.63.2 Syntax*

**Descriptive**

MAXRATES [*interval*]

**Diagram**




### *7.63.3 Parameter*

**interval** The new interval time in minutes. The default value is 1440 minutes (1 day).

### *7.63.4 Examples*

**Example 1:**

Change the MAXRATES interval from the default value (or the last manually set interval value) to 60 minutes.

```
2007/06/30 14:24:59.343 00000FC4 maxrates 60
2007/06/30 14:24:59.343 00000FC4 Maxrates interval set to 60 minutes.
```

**Figure 106: MAXRATES command (set the interval time)**

**Example 2:**

Display the maximum observed MIPS and I/O rates during the last interval.

```
2007/06/30 14:24:54.262 00000FC4 maxrates
2007/06/30 14:24:54.262 00000FC4 Highest observed MIPS/SIOS rates:
2007/06/30 14:24:54.262 00000FC4
2007/06/30 14:24:54.262 00000FC4   From: Wed Sep 20 14:13:20 2005
2007/06/30 14:24:54.262 00000FC4   To:   Wed Sep 20 14:13:20 2005
2007/06/30 14:24:54.262 00000FC4
2007/06/30 14:24:54.262 00000FC4         MIPS: 0.00
2007/06/30 14:24:54.262 00000FC4         SIOS: 0
2007/06/30 14:24:54.262 00000FC4
2007/06/30 14:24:54.262 00000FC4   From: Wed Sep 20 14:13:20 2005
2007/06/30 14:24:54.262 00000FC4   To:   Wed Sep 20 14:24:54 2005
2007/06/30 14:24:54.262 00000FC4
2007/06/30 14:24:54.262 00000FC4         MIPS: 397.425046
2007/06/30 14:24:54.262 00000FC4         SIOS: 5441
2007/06/30 14:24:54.262 00000FC4
2007/06/30 14:24:54.262 00000FC4 Current interval = 1440 minutes.
```

**Figure 107: MAXRATES command (display maximum rates)**

## 7.64 MESSAGE (Display message on console like VM)

### *7.64.1 Function*

The MESSAGE command is actually a VM CP command. It is used to transmit message text to the virtual console of other active users.

However, some programmers also use this command as a simplified way to display messages on the own virtual machine console by using "MESSAGE * <message text>". One program that uses this is the IBM "System z" port of Solaris which is only capable of running under z/VM.

So basically, all the MESSAGE command does is to display a message on the Hercules console with VM like headers. Therefore the MESSAGE command is not really intended to be used directly from the Hercules console as shown in the example below, although it does not hurt doing so.

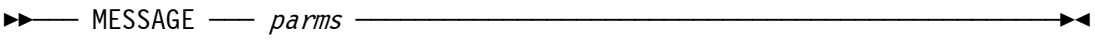
MSG is the same as MESSAGE. See also the MSG and MSGNOH commands in this manual.

### *7.64.2 Syntax*

**Descriptive**

MESSAGE *parms*

**Diagram**




### *7.64.3 Parameter*

**parms** *parms* specifies the required and optional parameters as described in IBM's VM "CP Commands and Utilities Reference" manual.

### *7.64.4 Examples*

**Example 1:**

Display a message on the own virtual machine console.

```
18:23:58 message * transmitted message to VM console
18:23:58  18:23:58  * MSG FROM HERCULES: transmitted message to VM console
```

**Figure 108: MESSAGE command**

## 7.65 MOUNTED_TAPE_REINIT (Control tape initialization)

### *7.65.1 Function*

The MOUNTED_TAPE_REINIT command controls whether reinitialization of tape drive devices via the DEVINIT command in order to mount a new tape should be allowed if there is already a tape mounted on the drive. Without a parameter given the current setting is displayed.

Specifiying ALLOW indicates that new tapes may be mounted via "DEVINIT nnnn new-tape-filename" irrespective of whether or not there is already a tape mounted on the drive.

Specifying DISALLOW prevents new tapes from being mounted if one is already mounted. Before the new tape can be mounted the existing one has first to be unmounted (via the "DEVINIT nnnn *" command). Otherwise the DEVINIT attempt to mount the new tape is rejected.

This option is meant as a safety mechanism to protect against accidentally unmounting a tape from the wrong device as a result of a simple typing error and thereby eventually cancelling a potentially important tape job.

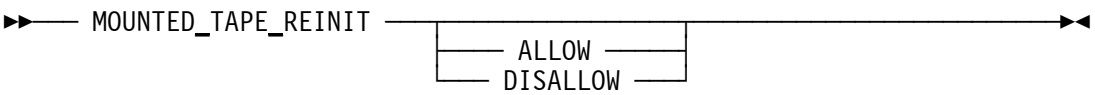
Please note that for SCSI tape drives the "DEVINIT nnnn *" command has no effect. The tape must be unmounted manually since it is a real physical device and not emulated via a disk file like '.AWS' or '.HET' tapes.

### *7.65.2 Syntax*

**Descriptive**

```
MOUNTED_TAPE_REINIT [ALLOW | DISALLOW]
```

**Diagram**




### *7.65.3 Parameter*

**ALLOW** Indicates that new tapes may be mounted irrespective of wether or not there is already a tape mounted on the drive. This is the default.

**DISALLOW** Prevents new tapes from being mounted if one is already mounted on the drive. Before the new tape can be mounted the currently mounted tape must first to be unmounted.

## *7.65.4 Examples*

**Example 1:**

Display current tape mount settings.

```
2009/03/07 22:20:01.328 00000D44 mounted_tape_reinit
2009/03/07 22:20:01.328 00000D44 Tape mount reinit allowed
```

**Figure 109: MOUNTED_TAPE_REINIT command (display settings).**

**Example 2:**

Disallow tape mount reinitialization.

```
2009/03/07 22:20:14.718 00000D44 mounted_tape_reinit disallow
2009/03/07 22:20:17.968 00000D44 mounted_tape_reinit
2009/03/07 22:20:17.968 00000D44 Tape mount reinit disallowed
```

**Figure 110: MOUNTED_TAPE_REINIT command (disallow tape mount reinitialization)**

## 7.66 MSG (Display message on console like VM)

### *7.66.1 Function*

The MSG command is actually a VM CP command. It is used to transmit message text to the virtual console of other active users.

However, some programmers also use this command as a simplified way to display messages on the own virtual machine console by using "MSG * <message text>". One program that uses this is the IBM "System z" port of Solaris which is only capable of running under z/VM.

So basically, all the MSG command does is to display a message on the Hercules console with VM like headers. Therefore the MSG command is not really intended to be used directly from the Hercules console as shown in the example below, although it does not hurt doing so.

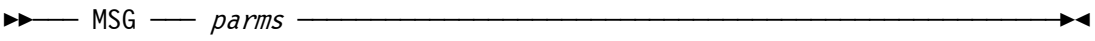
MESSAGE is the same as MSG. See also the MESSAGE and MSGNOH commands in this manual.

### *7.66.2 Syntax*

**Descriptive**

MSG *parms*

**Diagram**




### *7.66.3 Parameter*

**parms** *parms* specifies the required and optional parameters as described in IBM's VM "CP Commands and Utilities Reference" manual.

### *7.66.4 Examples*

**Example 1:**

Display a message on the own virtual machine console.

```
18:24:09 msg * transmitted message to VM console
18:24:09  18:24:09  * MSG FROM HERCULES: transmitted message to VM console
```

**Figure 111: MSG command**

## 7.67 MSGHLD (Display or set timeout of held messages)

### *7.67.1 Function*

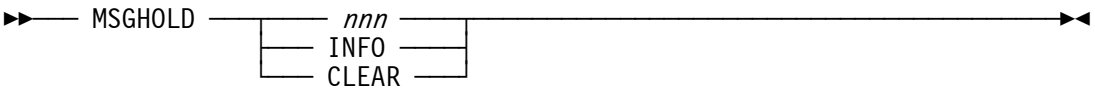
The MSGHLD console command is used to display or set the timeout value of held messages. It is also possible to release all held messages without having to wait for the timeout period to expire.

### *7.67.2 Syntax*

**Descriptive**

```
MSGHLD {nnn | INFO | CLEAR}
```

**Diagram**




### *7.67.3 Parameter*

**nnn** This value specifies the new timeout value of held messages in seconds.

**INFO** Display the current message held time setting.

**CLEAR** Release the held messages immediately.

### *7.67.4 Examples*

**Example 1:**

Set the new timeout interval for held messages to 30 seconds.

```
2009/01/08 09:09:34.781 00000CA0 msghld 30
2009/01/08 09:09:34.781 00000CA0 Current message held time is set to 30 seconds.
```

**Figure 112: MSGHELD command (set new interval)**

**Example 2:**

Release all held messages.

```
2009/01/08 09:10:38.241 00000CA0 msghld clear
2009/01/08 09:10:38.241 00000CA0 Held messages cleared
```

**Figure 113: MSGHELD command (release messages)**

## 7.68 MSGNOH (Display message on console like VM, but without header)

### 7.68.1 Function

The MSGNOH command is actually a VM CP command. It is used to transmit message text to the virtual console of other active users without the standard MESSAGE command header. However, some programmers also use this command as a simplified way to display messages on the own virtual machine console by using "MSGNOH * <message text>".

So basically, all the MSGNOH command does is to display a message on the Hercules console (like the MESSAGE and MSG commands) but without VM like headers. Therefore the MSGNOH command is not really intended to be used directly from the Hercules console as shown in the example below, although it does not hurt doing so.

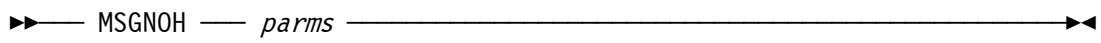
See also the MESSAGE and MSG commands in this manual.

### 7.68.2 Syntax

**Descriptive**

MSGNOH *parms*

**Diagram**




### 7.68.3 Parameter

**parms** *parms* specifies the required and optional parameters as described in IBM's VM "CP Commands and Utilities Reference" manual.

### 7.68.4 Examples

**Example 1:**

Display a message on the own virtual machine console.

```
18:24:19 msgnoh * transmitted message to VM console
18:24:19 transmitted message to VM console
```

**Figure 114: MSGNOH command**

## 7.69 OSTAILOR (Specify intended operating system)

### *7.69.1 Function*

The OSTAILOR command lets you display or (re-)specify the intended operating system. The effect of this parameter is to reduce control panel message traffic by selectively suppressing trace messages for program checks which are considered normal in the specified environment.

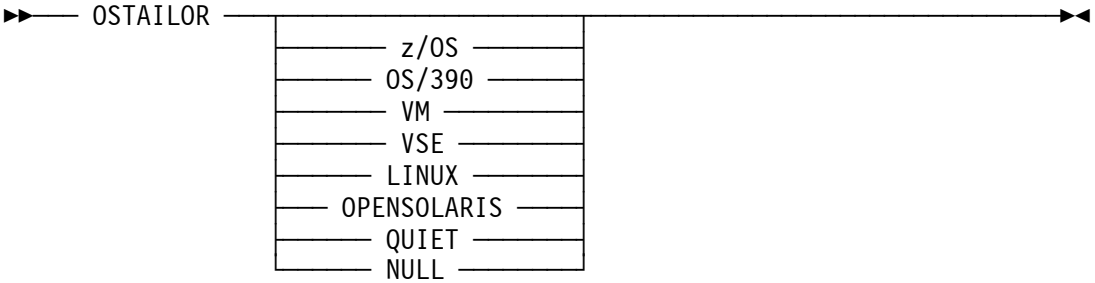
The argument *QUIET* suppresses all exception messages, whereas the argument *NULL* suppresses none of them. The other options do suppress some messages and do not supress other messages depending on the specified operating system.

### *7.69.2 Syntax*

**Descriptive**

```
OSTAILOR [z/OS | OS/390 | VM | VSE | LINUX | OPENSOLARIS | QUIET | NULL]
```

**Diagram**




### *7.69.3 Parameter*

| | |
|---|---|
| **z/OS** | Code z/OS if you intend to run z/OS. |
| **OS/390** | Code OS/390 if you intend to run MVS/370, MVS/XA, MVS/ESA, OS/390. |
| **VM** | Code VM if you intend to run VM/370, VM/ESA or z/VM. |
| **VSE** | Code VSE if you intend to run VSE/370, VSE/ESA or z/VSE. |
| **LINUX** | Code Linux you intend to run Linux/390 or Linux for z/Series. |
| **OpenSolaris** | Code OpenSolaris you intend to run OpenSolaris for z/Series. |
| **QUIET** | QUIET discards all exception messages. |
| **NULL** | NULL allows all exception messages to be logged. |

### *7.69.4 Examples*

**Example 1:**

Display the currently specified intended operating system.

```
2007/06/30 15:11:36.640 00000FC4 ostailor
2007/06/30 15:11:36.671 00000FC4 OSTAILOR z/OS
```

**Figure 115: OSTAILOR command (display intended operating system)**

**Example 2:**

Change the specified intended operating system.

```
2007/06/30 15:11:48.187 00000FC4 ostailor linux
2007/06/30 15:11:51.890 00000FC4 ostailor
2007/06/30 15:11:51.890 00000FC4 OSTAILOR LINUX
```

**Figure 116: OSTAILOR command (specify intended operating system)**

## 7.70 PANRATE (Display or set console refresh rate)

### 7.70.1 Function

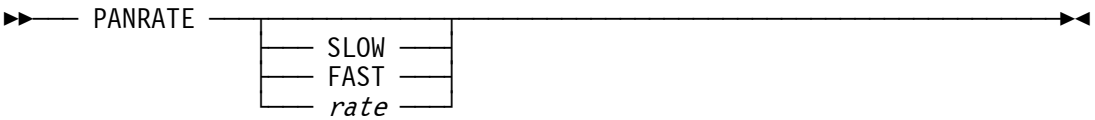
The PANRATE command shows the current setting of the console (panel) refresh rate and allows a new value to be set.

### 7.70.2 Syntax

**Descriptive**

PANRATE [SLOW | FAST | *rate*]

**Diagram**




### 7.70.3 Parameter

**SLOW** SLOW is a synonym for a panel refresh rate of 500 milliseconds.

**FAST** FAST is a synonym for a panel refresh rate of 50 milliseconds.

**rate** Any value between 1 (10) and 5000 milliseconds. A value less than the Linux system clock tick interval (10 on Intel platforms, 1 on Alpha platforms) or a value of more than 5000 will be rejected.

### 7.70.4 Examples

**Example 1:**

Display current panel refresh rate.

```
2007/06/30 14:31:53.653 00000FC4 panrate
2007/06/30 14:31:53.653 00000FC4 HHCPN037I Panel refresh rate = 1000 millisecond(s)
```

**Figure 117: PANRATE command (list current value)**

**Example 2:**

Set a new panel refresh rate.

```
2007/06/30 14:31:55.623 00000FC4 panrate fast
2007/06/30 14:31:55.623 00000FC4 HHCPN037I Panel refresh rate = 50 millisecond(s)
```

**Figure 118: PANRATE command (set new value)**

## 7.71 PGMTRACE (Trace program interrupts)

### *7.71.1 Function*

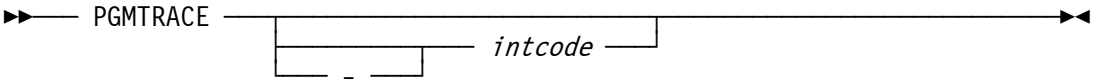
The PGMTRACE command without arguments displays the actual trace program interrupt options. Given with an argument (interruption code) the command changes the current options. Precede the interruption code with a minus sign to stop tracing of that particular interruption code.

### *7.71.2 Syntax*

**Descriptive**

PGMTRACE [[-]*intcode*]

**Diagram**




### *7.71.3 Parameter*

**intcode** *intcode* specifies the interruption code for which trace information is to be written to the Hercules log or for which the trace information is to be stopped. *Intcode* has to be a valid program interruption code in the range from 0x01 to 0x40.

**-** A minus sign, directly preceeding the *intcode* parameter, stops tracing for that particular interruption code.

### *7.71.4 Examples*

**Example 1:**

Display the current program interruption trace settings.

```
12:24:28.718 00000FC4 pgmtrace
12:24:28.718 00000FC4  * = Tracing suppressed; otherwise tracing enabled
12:24:28.718 00000FC4  00000000000000001111111111111111222222222222222233333333333333334
12:24:28.718 00000FC4  123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0
12:24:28.718 00000FC4     *  *        **    *     *              **   **  *   *  *    *
```

**Figure 119: PGMTRACE command (display settings)**

**Example 2:**

Change the current program interruption trace settings.

```
12:24:37.781 00000FC4 pgmtrace 04
12:24:41.265 00000FC4 pgmtrace
12:24:41.265 00000FC4  * = Tracing suppressed; otherwise tracing enabled
12:24:41.265 00000FC4  00000000000000011111111111111112222222222222222333333333333333334
12:24:41.265 00000FC4  123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0
12:24:41.265 00000FC4         *        **    *     *              **   **  *   *  *    *
```

**Figure 120: PGMTRACE command (change settings)**

## 7.72 PR (Display prefix register)

### *7.72.1 Function*

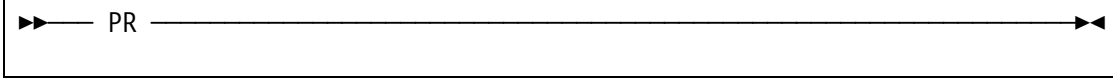
The PR command displays the contents of the prefix registers.

### *7.72.2 Syntax*

**Descriptive**

PR

**Diagram**




### *7.72.3 Parameter*

None.

### *7.72.4 Examples*

**Example 1:**

Display the prefix register.

```
2007/06/30 14:25:13.997 00000FC4 pr
2007/06/30 14:25:13.997 00000FC4 Prefix=3FBE3000
```

**Figure 121: PR command**

## 7.73 PSCP (Send system control program priority message)

### *7.73.1 Function*

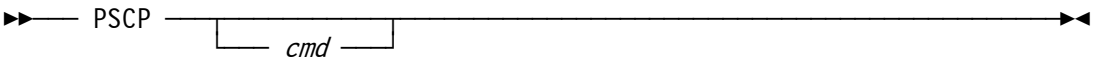
The PSCP command sends a system control program (i.e. guest operating system) priority command in any CMDTGT mode.

### *7.73.2 Syntax*

**Descriptive**

PSCP [*cmd*]

**Diagram**




### *7.73.3 Parameter*

**cmd** This is the command to be sent as a system control program (i.e. guest operating system) priority command.

### *7.73.4 Examples*

For a similar sample please refer to the SCP command.

## 7.74 PSW (Display or alter program status word)

### *7.74.1 Function*

The PSW command displays or alters the actual content of the Program Status Word (PSW).

### *7.74.2 Syntax*

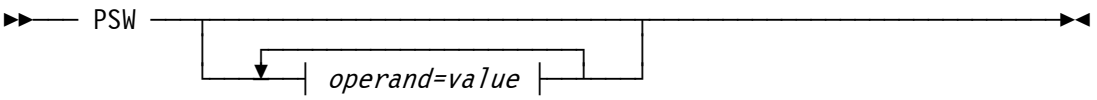
**Descriptive**

PSW [*operand=value* [*operand=value* ... ]]

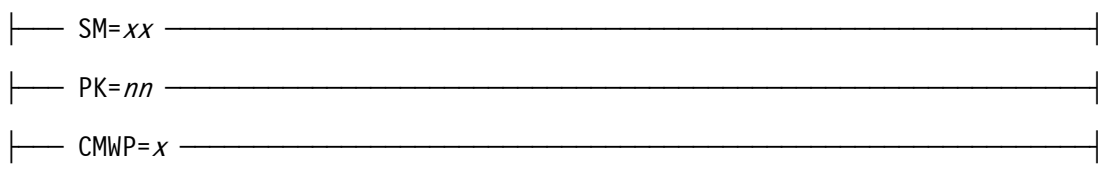
where operand is one of the following:

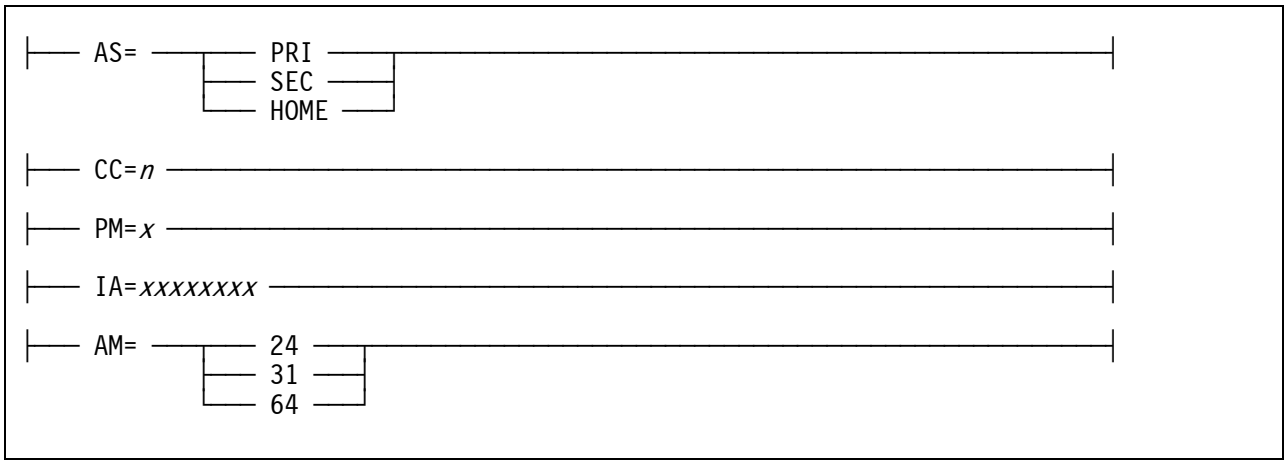
SM=*xx*
PK=*nn*
CMWP=*x*
AS=[PRI | SEC | HOME]
CC=*n*
PM=*x*
IA=*xxxxxxxx*
AM=[24 | 31 | 64]

**Diagram**




where operand is one of the following:






## 7.74.3 Parameter

| | |
|---|---|
| **SM=xx** | Modifies the PSW system mask (xx is 2 hexadecimal digits). |
| **PK=nn** | Modifies the PSW protection key (nn is decimal 0 to 15). |
| **CMWP=x** | Modifies the EC/M/W/P bits of the PSW (x is 1 headecimal digit). |
| **AS=[ pri \| sec \| home ]** | This modifies the PSW address space control bits. |
| **CC=n** | Modifies the PSW condition code (n is decimal 0 to 3). |
| **PM=x** | Modifies the PSW program mask (x is one hexadecimal digit). |
| **IA=xxxxxxxx** | Modifies the PSW instruction address (xxxxxxxx is one to 16 hexa-decimal digits). |
| **AM=[ 24 \| 31 \| 64 ]** | Modifies the addressing mode bits of the PSW. |

## 7.74.4 Examples

**Example 1:**

Display the Program Status Word.

```
2007/12/07 07:04:31.609 00000DBC psw
2007/12/07 07:04:31.609 00000DBC psw sm=04 pk=0 cmwp=C as=ar cc=2 pm=0 am=31 ia=15E43D8
2007/12/07 07:04:31.609 00000DBC PSW=040C6000 815E43EA
```

**Figure 122: PSW command (display PSW).**

**Example 2:**

Change the PSW condition code to zero.

```
2007/12/07 07:05:09.685 00000DBC psw cc=0
2007/12/07 07:05:09.685 00000DBC psw sm=04 pk=0 cmwp=C as=ar cc=0 pm=0 am=31 ia=15E43D8
2007/12/07 07:05:09.685 00000DBC PSW=040C4000 815E43D8
```

**Figure 123: PSW command (modify PSW)**

**Example 3:**

Change the PSW condition code to zero and the addressing mode bits to AM=24.

```
2007/12/07 07:10:49.006 00000DBC psw cc=0 am=24
2007/12/07 07:10:49.006 00000DBC psw sm=04 pk=0 cmwp=C as=ar cc=0 pm=0 am=24 ia=15E43D8
2007/12/07 07:10:49.006 00000DBC PSW=040C4000 815E43EA
```

**Figure 124: PSW command (modify PSW)**

## 7.75 PTT (Set / display pthread trace)

### *7.75.1 Function*

The PTT command sets or displays the internal trace options. When specified with no operands, the PTT command displays the trace options and the contents of the internal trace table.

When specified with operands, the PTT command sets the trace options and/or specifies which events are to be traced. If the last operand is numeric, it sets the size of the trace table and activates the trace.

The following events can be traced:

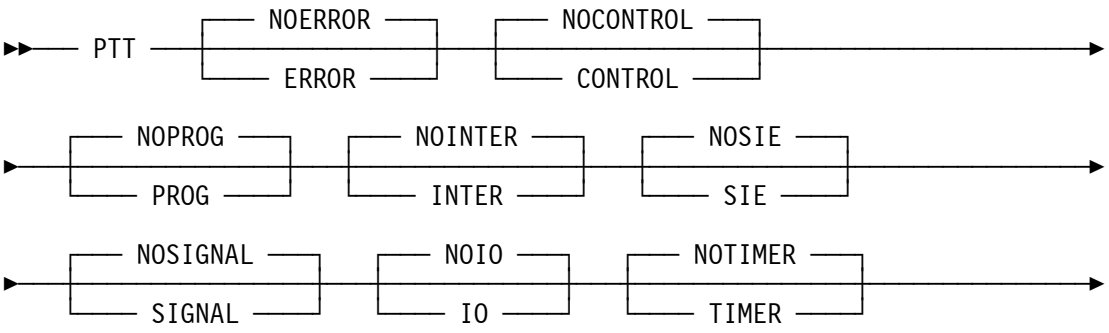
- error trace
- control trace
- program interrupt trace
- interlock failure trace
- sie trace
- signaling trace
- i/o trace
- timer trace
- thread trace
- logger trace
- lock trace buffer
- timestamp trace entries

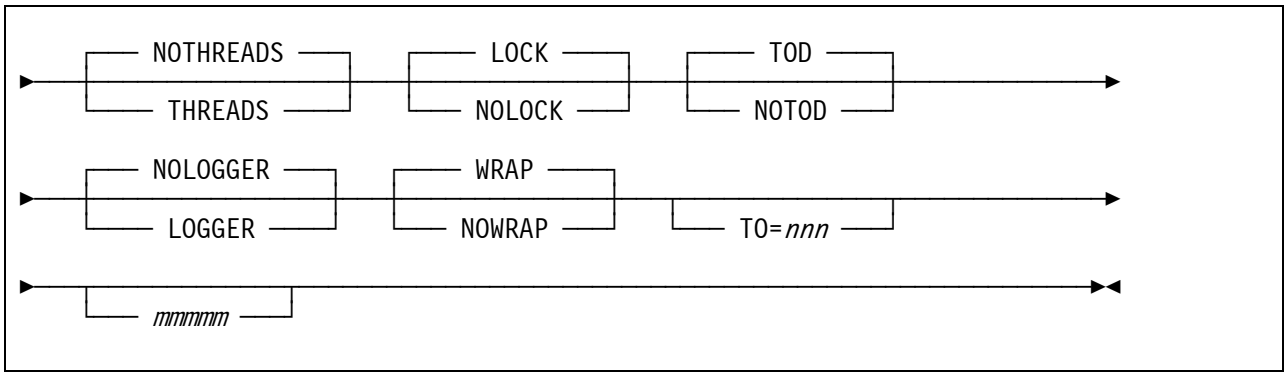
### *7.75.2 Syntax*

**Descriptive**

```
PTT [NOERROR | ERROR] [NOCONTROL | CONTROL] [NOPROG | PROG] [NOINTER | INTER]
    [NOSIE | SIE] [NOSIGNAL | SIGNAL] [NOIO | IO] [NOTIMER | TIMER]
    [NOTHREADS | THREADS] [NOLOCK | LOCK] [NOTOD | TOD] [NOLOGGER | LOGGER]
    [NOWRAP | WRAP] [TO=nnn] [mmmmm]
```

**Diagram**






## 7.75.3 Parameter

**NOxxxxxxxx** Deselects the specified trace entry type or deselects the specified option.

**ERROR** Selects the error trace entries.

**CONTROL** Selects the control trace entries.

**PROG** Selects the program interrupt trace entries.

**INTER** Selects the interlock failure trace entries.

**SIE** Selects the sie trace entries.

**SIGNAL** Selects the signaling trace entries.

**IO** Select the i/o trace entries.

**TIMER** Selects the timer trace entries.

**THREADS** Selects the thread trace entries.

**LOCK** Selects the lock trace buffer entries.

**TOD** Selects the timestamp trace entries.

**LOGGER** Selects the logger trace entries.

**WRAP** Specifies that the trace table has be overwritten when it is full. As soon as the end of the table is reached, the next entry to be written to the table goes to the start and overlays the trace entry that was formerly there.

**TO=nnn** Timeout in seconds before the automatic issuance of the PTT command is to occur, to print (display) the internal trace table.

**mmmmm** Number of trace entries to be kept.

## 7.75.4 Examples

**Example 1:**

Display the current defaults, then set the pthread trace options and start the trace.

```
2009/03/13 10:33:33.359 000003BC ptt
2009/03/13 10:33:33.359 000003BC HHCPT003I ptt lock tod wrap to=0 0
2009/03/13 10:33:33.359 000003BC ptt prog inter signal io threads logger nowrap 10000
```

**Figure 125: PTT command (display trace options / set options and start trace)**

**Example 2:**

Display the pthread trace entries.

```
14:04:35 ptt
14:04:35 00000ed0 lock before  007c3684 00000000 control.c:5483 14:04:07.509191
14:04:35 00000ed0 lock after   007c3684 00000000 control.c:5483 14:04:07.509201 0
14:04:35 00000ed0 unlock       007c3684 00000000 control.c:5483 14:04:07.509206 0
14:04:35 00000ed0 STSCH        00010013 000320c0 io.c:664       14:04:07.509496 00000ac4
14:04:35 00000ed0 lock before  012c0c88 00000000 io.c:689       14:04:07.509524
14:04:35 00000ed0 lock after   012c0c88 00000000 io.c:689       14:04:07.509526 0
14:04:35 00000ed0 unlock       012c0c88 00000000 io.c:694       14:04:07.509528 0
14:04:35 00000ed0 SSCH         00010013 00000470 io.c:516       14:04:07.509543 00000ac4
14:04:35 00000ed0 lock before  012c0c88 00000000 channel.c:2199 14:04:07.509546
14:04:35 00000ed0 lock after   012c0c88 00000000 channel.c:2199 14:04:07.509548 0
14:04:35 00000ed0 unlock       012c0c88 00000000 channel.c:2297 14:04:07.509550 0
14:04:35 00000ed0 lock before  012c0c88 00000000 channel.c:2430 14:04:07.509576
14:04:35 00000ed0 lock after   012c0c88 00000000 channel.c:2430 14:04:07.509578 0
14:04:35 00000ed0 unlock       012c0c88 00000000 channel.c:2460 14:04:07.509580 0
14:04:35 00000ed0 lock before  012c194c 00000000 cckddasd.c:479 14:04:07.509584
14:04:35 00000ed0 lock after   012c194c 00000000 cckddasd.c:479 14:04:07.509586 0
14:04:35 00000ed0 lock before  004bd998 00000000 cache.c:141    14:04:07.509588
14:04:35 00000ed0 lock after   004bd998 00000000 cache.c:141    14:04:07.509589 0
14:04:35 00000ed0 unlock       004bd998 00000000 cache.c:148    14:04:07.509607 0
14:04:35 00000ed0 unlock       012c194c 00000000 cckddasd.c:519 14:04:07.509612 0
14:04:35 00000ed0 lock before  012c194c 00000000 cckddasd.c:546 14:04:07.509623
14:04:35 00000ed0 lock after   012c194c 00000000 cckddasd.c:546 14:04:07.509625 0
14:04:35 00000ed0 lock before  004bd998 00000000 cache.c:141    14:04:07.509627
14:04:35 00000ed0 lock after   004bd998 00000000 cache.c:141    14:04:07.509629 0
14:04:35 00000ed0 unlock       004bd998 00000000 cache.c:148    14:04:07.509631 0
14:04:35 00000ed0 unlock       012c194c 00000000 cckddasd.c:564 14:04:07.509632 0
14:04:35 00000ed0 lock before  012c0c88 00000000 channel.c:3269 14:04:07.509634
14:04:35 00000ed0 lock after   012c0c88 00000000 channel.c:3269 14:04:07.509636 0
14:04:35 00000ed0 lock before  007c368c 00000000 channel.c:3333 14:04:07.509638
14:04:35 00000ed0 lock after   007c368c 00000000 channel.c:3333 14:04:07.509639 0
14:04:35 00000ed0 unlock       007c368c 00000000 channel.c:3333 14:04:07.509641 0
14:04:35 00000ed0 unlock       012c0c88 00000000 channel.c:3335 14:04:07.509642 0
14:04:35 00000ed0 lock before  007c3684 00000000 channel.c:3338 14:04:07.509644
14:04:35 00000ed0 lock after   007c3684 00000000 channel.c:3338 14:04:07.509646 0
14:04:35 00000ed0 lock before  007c368c 00000000 channel.c:3340 14:04:07.509647
14:04:35 00000ed0 lock after   007c368c 00000000 channel.c:3340 14:04:07.509649 0
14:04:35 00000ed0 unlock       007c368c 00000000 channel.c:3340 14:04:07.509651 0
14:04:35 00000ed0 unlock       007c3684 00000000 channel.c:3341 14:04:07.509652 0
14:04:35 00000ed0 lock before  007c3684 00000000 cpu.c:1555     14:04:07.509663
14:04:35 00000ed0 lock after   007c3684 00000000 cpu.c:1555     14:04:07.509665 0
14:04:35 00000ed0 lock before  007c368c 00000000 channel.c:3504 14:04:07.509666
14:04:35 00000ed0 lock after   007c368c 00000000 channel.c:3504 14:04:07.509668 0
14:04:35 00000ed0 unlock       007c368c 00000000 channel.c:3543 14:04:07.509677 0
14:04:35 00000ed0 lock before  012c0c88 00000000 channel.c:3555 14:04:07.509681
14:04:35 00000ed0 lock after   012c0c88 00000000 channel.c:3555 14:04:07.509683 0
14:04:35 00000ed0 lock before  007c368c 00000000 channel.c:3558 14:04:07.509684
14:04:35 00000ed0 lock after   007c368c 00000000 channel.c:3558 14:04:07.509694 0
14:04:35 00000ed0 unlock       007c368c 00000000 channel.c:3634 14:04:07.509697 0
14:04:35 00000ed0 unlock       012c0c88 00000000 channel.c:3635 14:04:07.509699 0
```

```
14:04:35 00000ed0 *IOINT       00010013 00000000 cpu.c:1159     14:04:07.509701 00000000
14:04:35 00000ed0 unlock       007c3684 00000000 cpu.c:1241     14:04:07.509704 0
14:04:35 00000ed0 STSCH        00010013 000320c0 io.c:664       14:04:07.509709 00000eb8
14:04:35 00000ed0 lock before  012c0c88 00000000 io.c:689       14:04:07.509711
14:04:35 00000ed0 lock after   012c0c88 00000000 io.c:689       14:04:07.509712 0
14:04:35 00000ed0 unlock       012c0c88 00000000 io.c:694       14:04:07.509714 0
14:04:35 00000ed0 TSCH         00010013 00032060 io.c:837       14:04:07.509716 00000eb8
14:04:35 00000ed0 lock before  012c0c88 00000000 channel.c:600  14:04:07.509724
14:04:35 00000ed0 lock after   012c0c88 00000000 channel.c:600  14:04:07.509726 0
14:04:35 00000ed0 lock before  007c368c 00000000 channel.c:799  14:04:07.509729
14:04:35 00000ed0 lock after   007c368c 00000000 channel.c:799  14:04:07.509731 0
14:04:35 00000ed0 unlock       007c368c 00000000 channel.c:799  14:04:07.509732 0
14:04:35 00000ed0 unlock       012c0c88 00000000 channel.c:801  14:04:07.509734 0
14:04:35 00000ed0 lock before  007c3684 00000000 channel.c:804  14:04:07.509736
14:04:35 00000ed0 lock after   007c3684 00000000 channel.c:804  14:04:07.509738 0
14:04:35 00000ed0 lock before  007c368c 00000000 channel.c:805  14:04:07.509740
14:04:35 00000ed0 lock after   007c368c 00000000 channel.c:805  14:04:07.509741 0
14:04:35 00000ed0 unlock       007c368c 00000000 channel.c:805  14:04:07.509743 0
14:04:35 00000ed0 unlock       007c3684 00000000 channel.c:806  14:04:07.509745 0
14:04:35 00000ed0 STSCH        00010013 000320c0 io.c:664       14:04:07.509779 00000ac4
14:04:35 00000ed0 lock before  012c0c88 00000000 io.c:689       14:04:07.509782
14:04:35 00000ed0 lock after   012c0c88 00000000 io.c:689       14:04:07.509784 0
14:04:35 00000ed0 unlock       012c0c88 00000000 io.c:694       14:04:07.509786 0
14:04:35 00000ed0 SSCH         00010013 00000450 io.c:516       14:04:07.509790 00000ac4
14:04:35 00000ed0 lock before  012c0c88 00000000 channel.c:2199 14:04:07.509792
14:04:35 00000ed0 lock after   012c0c88 00000000 channel.c:2199 14:04:07.509793 0
14:04:35 00000ed0 unlock       012c0c88 00000000 channel.c:2297 14:04:07.509795 0
14:04:35 00000ed0 lock before  012c0c88 00000000 channel.c:2430 14:04:07.509798
14:04:35 00000ed0 lock after   012c0c88 00000000 channel.c:2430 14:04:07.509799 0
14:04:35 00000ed0 unlock       012c0c88 00000000 channel.c:2460 14:04:07.509801 0
14:04:35 00000ed0 lock before  012c194c 00000000 cckddasd.c:479 14:04:07.509803
14:04:35 00000ed0 lock after   012c194c 00000000 cckddasd.c:479 14:04:07.509805 0
14:04:35 00000ed0 lock before  004bd998 00000000 cache.c:141    14:04:07.509806
.
.
.
```

**Figure 126: PTT command (display trace entries)**

**Example 3:**

Set the pthread trace options, start the trace and issue automatic display of the trace entries after 2 minutes.

```
2009/03/13 10:48:25.248 000003BC ptt prog inter signal io threads logger nowrap to=120 10000
```

**Figure 127: PTT command (start trace and issue automatic display)**

## 7.76 PWD (Print working directory)

### *7.76.1 Function*

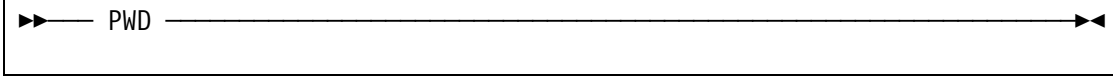
The PWD command prints the current working directory.

### *7.76.2 Syntax*

**Descriptive**

```
PWD
```

**Diagram**




### *7.76.3 Parameter*

None.

### *7.76.4 Examples*

**Example 1:**

Print the current working directory.

```
2007/08/23 00:50:46.859 00000B54 pwd
2007/08/23 00:50:46.859 00000B54 D:\Hercules
```

**Figure 128: PWD command**

## 7.77 QD (Query DASD)

### *7.77.1 Function*

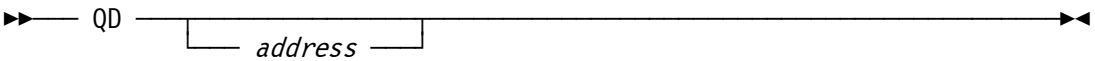
The QD command queries one or all (depending on the argument given) DASD devices.

### *7.77.2 Syntax*

**Descriptive**

QD [*address*]

**Diagram**




### *7.77.3 Parameter*

**address** The address of the DASD device to be queried. If no address is given, then a query for all DASD devices is listed.

### *7.77.4 Examples*

**Example 1:**

Query DASD device with address 0AE0.

```
02:55:40.954 000009EC qd 0ae0
02:55:40.954 000009EC 0ae0 SNSID 00  ff3990e9 33900a00 40fa0100 41270004
02:55:40.954 000009EC            10  423e0060
02:55:40.954 000009EC 0ae0 RDC   00  3990e933 900a5000 30972024 0d0b000f
02:55:40.954 000009EC            10  e000e5a2 05940222 13090674 00000000
02:55:40.954 000009EC            20  00000000 00000000 24241502 dfee0001
02:55:40.954 000009EC            30  0677080f 00004800 15ff0000 00000000
02:55:40.954 000009EC 0ae0 RCD   00  c4010100 4040f3f3 f9f0f0f0 c1c8d9c3 |D...  339000AHRC|
02:55:40.954 000009EC            10  e9e9f0f0 f0f0f0f0 f0f0f0f0 f0f10300 |ZZ000000000001..|
02:55:40.954 000009EC            20  c4000000 4040f3f3 f9f0f0f0 c1c8d9c3 |D...  339000AHRC|
02:55:40.954 000009EC            30  e9e9f0f0 f0f0f0f0 f0f0f0f0 f0f10300 |ZZ000000000001..|
02:55:40.954 000009EC            40  d4020000 4040f3f9 f9f0f0c5 f9c8d9c3 |M...  39900E9HRC|
02:55:40.954 000009EC            50  e9e9f0f0 f0f0f0f0 f0f0f0f0 f0f10300 |ZZ000000000001..|
02:55:40.954 000009EC            60  f0000001 4040f3f9 f9f04040 40c8d9c3 |0...  3990   HRC|
02:55:40.954 000009EC            70  e9e9f0f0 f0f0f0f0 f0f0f0f0 f0f10300 |ZZ000000000001..|
02:55:40.954 000009EC            80  00000000 00000000 00000000 00000000 |................|
```

```
02:55:40.954 000009EC            90  00000000 00000000 00000000 00000000 |................|
02:55:40.954 000009EC            a0  00000000 00000000 00000000 00000000 |................|
02:55:40.970 000009EC            b0  00000000 00000000 00000000 00000000 |................|
02:55:40.970 000009EC            c0  00000000 00000000 00000000 00000000 |................|
02:55:40.970 000009EC            d0  00000000 00000000 00000000 00000000 |................|
02:55:40.970 000009EC            e0  80000007 00001e00 0ae080e0 e0e00700 |........\.\\\..|
02:55:40.970 000009EC            f0  008080e0 00000000 00000000 00000000 |...\............|
02:55:40.970 000009EC 0ae0 SNSS  00  01e01f00 00000000 00000000 00000000
02:55:40.970 000009EC            10  00000000 00000000 00000000 00000000
02:55:40.970 000009EC            20  00000000 00000ae0 00000000
```

**Figure 129: QD command**

## *7.77.5 Explanations*

The Sense ID (SNSID) is describing the type and the model number of the subsystem and logical volume of the channel and has the following format:

| Bytes | Description |
|---|---|
| 0 | --- |
| 1-2 | Subsystem Type |
| 3 | Subsystem Model and Architecture |
| 4-5 | Device Type |
| 6 | Device Model |
| 7 | Reserved |
| 8-11 | Command Interface Word (CIW) for Read Configuration Data |
| 12-15 | Command Interface Word (CIW) for Set-Interface-Identifier |
| 16-19 | Command Interface Word (CIW) for Read-Node-Identifier |

**Table 12: Sense ID**

The Read Device Characteristics (RDC) defines the characteristics of the logical volume. The format is the following:

| Bytes | Description |
|---|---|
| 0-1 | Subsystem type |
| 2 | Subsystem model number and architecture |
| 3-4 | Device type |
| 5 | Device model |
| 6-9 | Subsystem and device facilities |
| 10 | Device class code |
| 11 | Device type code |
| 12-13 | Number of primary cylinders |
| 14-15 | Tracks per cylinder |
| 16 | Number of sectors |
| 17-19 | Track length |
| 20-21 | Length of HA and R0 |

| | |
|---|---|
| 22 | Track capacity calculation formula |
| 23-27 | Track capacity calculation factors F1 through F5 |
| 28-29 | First alternate cylinder address |
| 30-31 | Number of alternate tracks |
| 32-33 | First diagnostic cylinder address |
| 34-35 | Number of diagnostic tracks |
| 36-37 | First device support cylinder address |
| 38-39 | number of device support tracks |
| 40 | MDR record ID |
| 41 | OBR record ID |
| 42 | Control unit type code |
| 43 | Read trackset parameter length |
| 44-45 | Maximum record zero length |
| 46 | Reserved |
| 47 | Track set size |
| 48 | Track capacity calculation factor F6 |
| 49-50 | RPS sector calculation factors |
| 51-53 | Reserved |
| 54 | Device and control unit features |
| 55 | Reserved |
| 56 | Real control unit type code |
| 57 | Real device type code |
| 58-63 | Reserved |

**Table 13: Read Device Characteristics**

The Read Configuration Data (RCD) contains configuration data. It shows how the internal disk subsystem is configured. The RCD has the following format:

| **Bytes** | **Name** | **Description** |
|---|---|---|
| 000-031 | NED1 | Volume NED (logical volume within the subsystem) |
| 032-063 | NED2 | NED for the RAID array |
| 064-095 | NED3 | NED for the logical subsystem |
| 096-127 | NED4 | Token NED (establishes a relationship among all configu-ration data records in a control unit in the subsystem) |
| 128-223 | | Reserved |
| 224-255 | QNEQ | General NEQ (describes paths and addresses) |

**Table 14: Read Configuration Data**

Legend:

NED Node Element Descriptor

NEQ Node Element Qualifier

## 7.78 QUIET (Toggle automatic refresh of console display data)

### *7.78.1 Function*

The QUIET command either disables automatic screen refreshing if it is currently enabled or enables it if it is currently disabled. When disabled you will not be able to see the response of any entered commands nor any messages issued by the system nor be able to scroll the display, etc. Basically all screen updating is disabled. Entering 'quiet' again re-enables screen updating.

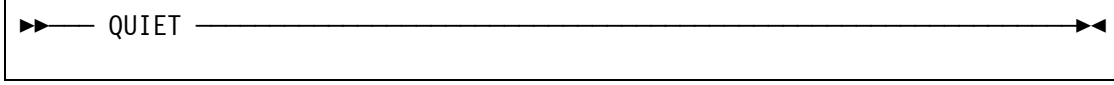
Please note, that the QUIET command is disabled in the dyngui and therefore is not available when using the Hercules WinGUI.

### *7.78.2 Syntax*

**Descriptive**

```
QUIET
```

**Diagram**




### *7.78.3 Parameter*

None.

### *7.78.4 Examples*

**Example 1:**

Change between automatic refresh of console display data.

```
2007/06/30 13:30:43.062 00000618 quiet
2007/06/30 13:30:43.062 00000618 HHCPN027I Automatic refresh disabled.
2007/06/30 13:30:47.906 00000618 quiet
2007/06/30 13:30:47.906 00000618 HHCPN027I Automatic refresh enabled.
```

**Figure 130: QUIET command**

## 7.79 QUIT (Terminate the emulator)

### *7.79.1 Function*

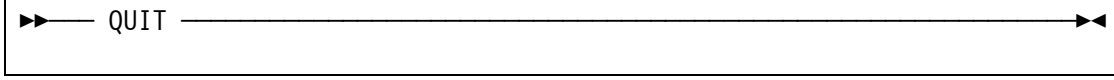
The QUIT command (see also the EXIT command) initiates the Hercules shutdown. It terminates all threads, detaches all channels and devices and releases the configuration. Finally it terminates the emulator.

### *7.79.2 Syntax*

**Descriptive**

```
QUIT
```

**Diagram**




### *7.79.3 Parameter*

None.

### *7.79.4 Examples*

**Example 1:**

Initiate the Hercules shutdown.

```
2007/06/30 2007/06/30 14:49:57.044 00000FC4 quit
2007/06/30 2007/06/30 14:49:57.044 00000FC4 HHCIN900I Begin Hercules shutdown
2007/06/30 2007/06/30 14:49:57.044 00000FC4 HHCIN901I Releasing configuration
2007/06/30 2007/06/30 14:49:57.044 00000FC4 HHCCP008I CPU0000 thread ended: tid=0000049C,pid=232
2007/06/30 2007/06/30 14:49:57.044 00000FC4 HHCCP008I CPU0001 thread ended: tid=0000046C,pid=232
2007/06/30 2007/06/30 14:49:57.044 00000FC4 HHCCF047I Subchannel 0000 detached
2007/06/30 2007/06/30 14:49:57.044 00000FC4 HHCCF047I Subchannel 0001 detached
2007/06/30 2007/06/30 14:49:57.044 00000FC4 HHCCF047I Subchannel 0002 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCTT003I Timer thread ended
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0003 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0004 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0005 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0006 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0007 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0008 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0009 detached
```

```
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 000A detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 000B detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 000C detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 000D detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 000E detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 000F detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0010 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0011 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0012 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0013 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCTE004I Console connection thread terminated
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCDA023I 0300 cache hits 812, misses 114, waits 0
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0014 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCDA023I 0310 cache hits 5, misses 7, waits 0
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0015 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCDA023I 0320 cache hits 271, misses 35, waits 0
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0016 detached
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCDA023I 0330 cache hits 137, misses 59, waits 0
2007/06/30 2007/06/30 14:49:57.059 00000FC4 HHCCF047I Subchannel 0017 detached
.
.
.
2007/06/30 2007/06/30 14:49:57.075 00000FC4 HHCCF047I Subchannel 0028 detached
2007/06/30 2007/06/30 14:49:57.075 00000FC4 HHCCF047I Subchannel 0029 detached
2007/06/30 2007/06/30 14:49:57.075 00000FC4 HHCCF047I Subchannel 002A detached
2007/06/30 2007/06/30 14:49:57.075 00000FC4 HHCCF047I Subchannel 002B detached
2007/06/30 2007/06/30 14:49:57.075 00000FC4 HHCCF047I Subchannel 002C detached
2007/06/30 2007/06/30 14:49:57.075 00000FC4 HHCCF047I Subchannel 002D detached
2007/06/30 2007/06/30 14:49:57.075 00000FC4 HHCCF047I Subchannel 002E detached
2007/06/30 2007/06/30 14:49:59.841 00000FC4 HHCCF047I Subchannel 002F detached
2007/06/30 2007/06/30 14:49:59.841 00000FC4 HHCIN902I Configuration release complete
2007/06/30 2007/06/30 14:49:59.841 00000FC4 HHCIN903I Calling termination routines
2007/06/30 2007/06/30 14:49:59.841 00000FC4 HHCHD900I Begin shutdown sequence
2007/06/30 2007/06/30 14:49:59.841 00000FC4 HHCHD901I Calling console_shutdown
2007/06/30 2007/06/30 14:49:59.841 00000FC4 HHCHD902I console_shutdown complete
2007/06/30 2007/06/30 14:49:59.841 00000FC4 HHCHD901I Calling hdl_term
2007/06/30 2007/06/30 14:49:59.841 00000FC4 HHCHD950I Begin HDL termination sequence
2007/06/30 2007/06/30 14:49:59.841 00000FC4 HHCHD951I Calling module dyngui cleanup routine
2007/06/30 2007/06/30 14:49:59.950 00000FC4 HHCHD952I Module dyngui cleanup complete
2007/06/30 2007/06/30 14:49:59.950 00000FC4 HHCHD951I Calling module *Hercules cleanup routine
2007/06/30 2007/06/30 14:49:59.950 00000FC4 HHCHD952I Module *Hercules cleanup complete
2007/06/30 2007/06/30 14:49:59.950 00000FC4 HHCHD959I HDL Termination sequence complete
2007/06/30 2007/06/30 14:49:59.950 00000FC4 HHCHD902I hdl_term complete
2007/06/30 2007/06/30 14:49:59.950 00000FC4 HHCHD901I Calling logger_term
2007/06/30 2007/06/30 14:49:59.950 00000FC4 HHCLG014I logger thread terminating
2007/06/30 2007/06/30 14:49:59.950 00000FC4 HHCLG014I logger thread terminating
2007/06/30 2007/06/30 14:49:59.950 00000FC4 HHCIN099I Hercules terminated
```

**Figure 131: QUIT command**

## 7.80 R (Display or alter real storage)

### *7.80.1 Function*

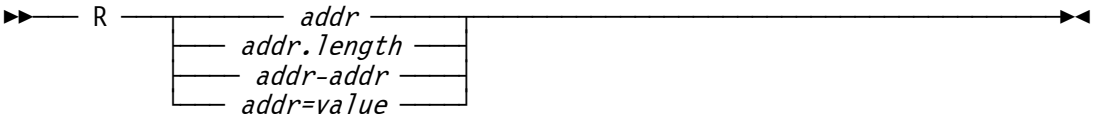
The R command allows you to display or alter real storage.

### *7.80.2 Syntax*

**Descriptive**

R {*addr* | *addr.length* | *addr-addr* | *addr=value*}

**Diagram**




### *7.80.3 Parameter*

**addr** Address of the real storage area that is to be displayed or altered.

**length** Length of the storage area that is to be displayed. The value of *.len* has to be given in hexadecimal. If the length is omitted then 64 bytes of real storage are displayed by default.

**value** Value is a hex-string of up to 32 pairs of digits (32 bytes) which will be written to the real storage address given by the *addr* parameter. After altering the storage, 16 bytes of real storage starting at *addr* are displayed.

### *7.80.4 Examples*

**Example 1:**

Display 256 bytes of real storage starting from location x'00000000'.

```
11:00:58.171 00000FC4 r 00000000.ff
11:00:58.171 00000FC4 R:00000000:K:0E=040C0000 81080278 00000000 00000000  ....a...........
11:00:58.171 00000FC4 R:00000010:K:0E=00FC7FC0 00000000 070C0000 81412E10  .."{........a...
11:00:58.171 00000FC4 R:00000020:K:0E=070C0000 80FD9AB0 070C2000 00047BB4  ..............#.
11:00:58.171 00000FC4 R:00000030:K:0E=00000000 00000000 070C2000 810C7C90  ............a.@.
11:00:58.171 00000FC4 R:00000040:K:0E=00000000 00000000 00000000 00FC7FC0  ..............."{
11:00:58.171 00000FC4 R:00000050:K:0E=00000000 00000000 040C0000 81398AE8  ............a..Y
11:00:58.171 00000FC4 R:00000060:K:0E=040C0000 80FFBA80 00080000 8F61D880  ............./Q.
```

```
11:00:58.171 00000FC4 R:00000070:K:0E=00080000 8F61E968 040C0000 813A7F80  ...../Z.....a.".
11:00:58.171 00000FC4 R:00000080:K:0E=00000000 00002401 00020078 00020011  ................
11:00:58.171 00000FC4 R:00000090:K:0E=0004B000 00000000 00000000 00000000  ................
11:00:58.171 00000FC4 R:000000A0:K:0E=00000000 013B8188 00000000 00000000  ......ah........
11:00:58.171 00000FC4 R:000000B0:K:0E=00000000 00000000 00010009 00F3CC08  ............3..
11:00:58.171 00000FC4 R:000000C0:K:0E=28000000 00000000 00000000 00000000  ................
11:00:58.171 00000FC4 R:000000D0:K:0E=00000000 0F5D2000 00000000 00000000  .....).........
11:00:58.171 00000FC4 R:000000E0:K:0E=00000000 00000000 00000000 00000000  ................
11:00:58.171 00000FC4 R:000000F0:K:0E=00000000 00000000 00000000 00000000  ................
```

**Figure 132: R command (display real storage)**

**Example 2:**

Alter 4 bytes of real storage at location x'00000000' to x'FFFFFFFF'.

```
11:01:55.015 00000FC4 r 00000000=ffffffff
11:01:55.015 00000FC4 R:00000000:K:0E=FFFFFFFF 81080278 00000000 00000000  ....a..........
```

**Figure 133: R command (alter real storage)**

## 7.81 RESTART (Generate restart interrupt)

### *7.81.1 Function*

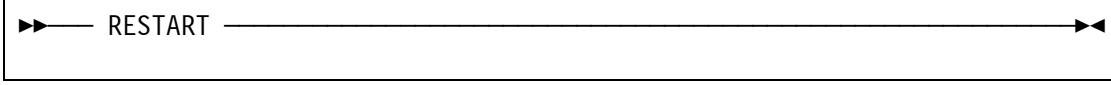
The RESTART command generates a restart interrupt (the virtual RESTART key is pressed).

### *7.81.2 Syntax*

**Descriptive**

RESTART

**Diagram**




### *7.81.3 Parameter*

None.

### *7.81.4 Examples*

**Example 1:**

Generate restart interrupt.

```
2007/06/30 15:30:23.703 00000FC4 restart
2007/06/30 15:30:23.718 00000FC4 HHCPN038I Restart key depressed
```

**Figure 134: RESTART command**

## 7.82 RESUME (Resume Hercules)

### *7.82.1 Function*

This command resumes a Hercules session that had been previously suspended with the SUSPEND command (see section 7.108 for details). The data necessary to resume the session is read from a packed (zipped) file called “hercules.srf.gz” located in the current configuration directory.

In order for an instance to be resumed HERCULES must be started with a configuration file describing the configuration at suspend time. For example, MAINSIZE and XPNDSIZE must match and all devices present at suspend time must be present at resume time.

Disk devices must be at the same state as they were at suspend time. They can however be a different file type. For example a disk could be a CCKD disk at suspend time then a CKD disk could be created using DASDCOPY and HERCULES resumed using the CKD disk instead.

HERCULES must also be configured similarly as at suspend time. For example if 4 emulated CPUs were active at suspend time then the session cannot be resumed on a HERCULES with a maximum of two CPUs. Similarly you will not be able to resume a session in z/Architecture mode for a HERCULES that was built without z/Architecture.

After entering the RESUME command on the Hercules console the suspend file is re-imported and the CPUs are put in the STARTED state again resuming guest program operations at the same point and in the same state as the suspend file was created.

There are some caveats when resuming guest operating system processing:

- As seen by the guest operating system, the TOD clock will appear to jump a large value. Some guests may not cope very well with this. For example some guests may be dismayed because certain interrupts will occur way past its due time. Also for S/370 an interval timer interrupt may be lost if the guest is interrupted for more than half the Interval Timer wrap time (around 8 hours).
- Although some effort has been put in order to make this as transparent as possible (that is, it should appear to the guest operating system that the STOP key was pressed for a large amount of time), some state information may be missed.

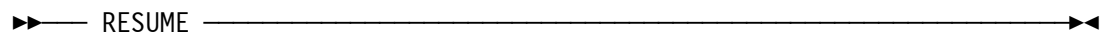
Some guest operating systems will fare better if the suspend state is prepared first. For MVS, as an example, it seems to help when a QUIESCE command and a SYSTEM RESTART manual operation are issued prior to suspend the system.

### *7.82.2 Syntax*

**Descriptive**

```
RESUME
```

**Diagram**



## 7.82.3 Parameter

None.

## 7.82.4 Examples

**Example 1:**

Resume a Hercules session.

```
14:16:29.921 00000FC4 resume
14:16:29.921 00000FC4 HHCCP008I CPU0000 thread ended: tid=00000C24, pid=4056
14:16:29.921 00000FC4 HHCCP008I CPU0001 thread ended: tid=0000044C, pid=4056
14:16:29.921 00000FC4 HHCSR001I Resuming suspended file created Wed Jan 11 14:15:43 2006
14:16:29.937 00000FC4 HHCTT003I Timer thread ended
14:16:31.187 00000FC4 HHCCP002I CPU0000 thread started: tid=00000EAC, pid=4056, priority=0
14:16:31.187 00000FC4 HHCCP003I CPU0000 architecture mode ESA/390
14:16:31.187 00000FC4 HHCTT002I Timer thread started: tid=00000760, pid=4056, priority=-20
14:16:31.187 00000FC4 HHCCP002I CPU0001 thread started: tid=00000ED8, pid=4056, priority=0
14:16:31.187 00000FC4 HHCCP003I CPU0001 architecture mode ESA/390
```

**Figure 135: RESUME command**

## 7.83 RMMOD (Delete a module)

### *7.83.1 Function*

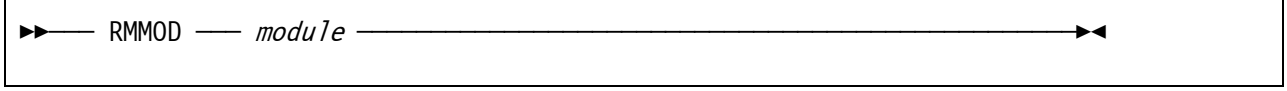
The RMMOD command deletes (unloads) a module.

### *7.83.2 Syntax*

**Descriptive**

RMMOD *module*

**Diagram**




### *7.83.3 Parameter*

**module** Name of the module that is to be unloaded. If a device is still bound to the module then a message is issued and the module is not deleted.

### *7.83.4 Examples*

**Example 1:**

Delete module “dyncrypt”.

```
2007/06/30 10:33:33.359 0000008E rmmod dyncrypt
2007/06/30 10:33:33.359 0000008E HHCHD102I Unloading dyncrypt ...
2007/06/30 10:33:33.359 0000008E HHCHD103I Module dyncrypt unloaded
```

**Figure 136: RMMOD command**

## 7.84 S (Instruction stepping)

### *7.84.1 Function*

This command sets the instruction stepping and breaking range. The instruction stepping is totally separate from the instruction tracing range (See T command).

A range can be specified. If there is no range then the range includes all addresses. “S 0” eliminates the range and all addresses will be stepped. With or without a given range, the S command displays whether instruction stepping is on or off and the range if any.

The S command by itself does not activate instruction stepping. Use the S+ command to activate instruction stepping.

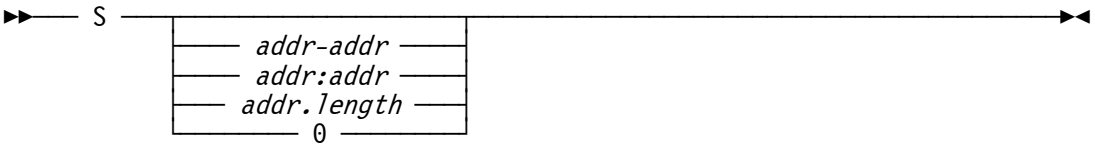
### *7.84.2 Syntax*

The S command has the following syntax:

**Descriptive**

S [*addr-addr* | *addr:addr* | *addr.length* | 0]

**Diagram**




### *7.84.3 Parameter*

**addr** Specifies an address. The range can be coded as “addr–addr” or “addr:addr” (from begin address to end address) or as address.length (from begin address for length).

**length** Specifies the length of the range.

**0** No range is specified or an existing range is reset. Instruction stepping is active for all addresses.

### *7.84.4 Examples*

**Example 1:**

Set instruction stepping range.

```
2009/02/15 10:06:22.432 00000FC4 s 1048576-1056768
2009/02/15 10:06:22.432 00000FC4 HHCPN040I Instruction stepping off range 1048576-1056768
```

**Figure 137: S command**

## 7.85 S+ (Instruction stepping on)

### *7.85.1 Function*

This command turns on the instruction stepping. After turning instruction stepping on, each ENTER from the console allows Hercules to execute exactly one instruction and lists detailed trace information.

The trace information includes the executing CPU, the PSW, the executed instruction in hexadecimal and in disassembled form as well as register hexadecimal displays of all involved register types.

A range can be specified as for the S command, otherwise the existing range is used. If there is no range (or range was specified as 0) then the range includes all addresses.

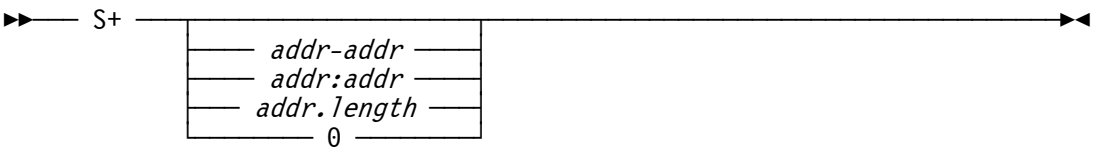
When an instruction within the range is about to be executed, the CPU is temporarily stopped and the next instruction is displayed. You may then examine registers and/or storage etc. before you press Enter to execute the instruction and stop at the next instruction. To turn off instruction stepping and continue normal execution enter the G command.

### *7.85.2 Syntax*

**Descriptive**

S+ [*addr-addr* | *addr:addr* | *addr.length* | 0]

**Diagram**




### *7.85.3 Parameter*

**addr** Specifies an address. The range can be coded as “addr–addr” or “addr:addr” (from begin address to end address) or as address.length (from begin address for length).

**length** Specifies the length of the range.

**0** No range is specified or an existing range is reset. Instruction stepping is active for all addresses.

## *7.85.4 Examples*

**Example 1:**

Turn on instruction stepping.

```
s+
HHCPN040I Instruction stepping on
CPU0000:  PSW=00082000 8000061E INST=D20700680240 MVC   104(8,0),576(0)        move_character
CPU0000:  R:00000068:K:06=00080000 8000060A 00000000 00000000  ................
CPU0000:  R:00000240:K:06=00080000 800009E6 00000000 80000000  .......W........
CPU0000: GR00=00000000  GR01=00000000  GR02=00000000  GR03=00000000
CPU0000: GR04=00000000  GR05=00000000  GR06=00000000  GR07=00000000
CPU0000: GR08=00000000  GR09=00000000  GR10=00000000  GR11=00000000
CPU0000: GR12=00000000  GR13=00000000  GR14=00000000  GR15=00000000

CPU0000:  PSW=00082000 80000624 INST=D20701D00248 MVC   464(8,0),584(0)        move_character
CPU0000:  R:000001D0:K:06=00000000 80000000 00000000 000009BE  ................
CPU0000:  R:00000248:K:06=00000000 80000000 00000000 000009BE  ................
CPU0000: GR00=00000000  GR01=00000000  GR02=00000000  GR03=00000000
CPU0000: GR04=00000000  GR05=00000000  GR06=00000000  GR07=00000000
CPU0000: GR08=00000000  GR09=00000000  GR10=00000000  GR11=00000000
CPU0000: GR12=00000000  GR13=00000000  GR14=00000000  GR15=00000000

CPU0000:  PSW=00082000 8000062A INST=1F55         SLR   5,5          subtract_logical_register
CPU0000: GR00=00000000  GR01=00000000  GR02=00000000  GR03=00000000
CPU0000: GR04=00000000  GR05=00000000  GR06=00000000  GR07=00000000
CPU0000: GR08=00000000  GR09=00000000  GR10=00000000  GR11=00000000
CPU0000: GR12=00000000  GR13=00000000  GR14=00000000  GR15=00000000

CPU0000:  PSW=00082000 8000062C INST=1F00         SLR   0,0          subtract_logical_register
CPU0000: GR00=00000000  GR01=00000000  GR02=00000000  GR03=00000000
CPU0000: GR04=00000000  GR05=00000000  GR06=00000000  GR07=00000000
CPU0000: GR08=00000000  GR09=00000000  GR10=00000000  GR11=00000000
CPU0000: GR12=00000000  GR13=00000000  GR14=00000000  GR15=00000000
.
.
.
```

**Figure 138: S+ command**

## 7.86 S- (Instruction stepping off)

### *7.86.1 Function*

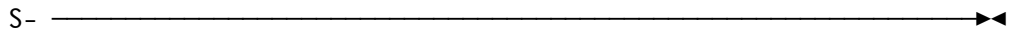
This command turns off the instruction stepping. To continue normal processing after stopping the instruction stepping use the G command.

### *7.86.2 Syntax*

**Descriptive**

S-

**Diagram**




### *7.86.3 Parameter*

None.

### *7.86.4 Examples*

**Example 1:**

Turn off instruction stepping.

```
2009/02/15 10:07:39.312 00000FC4 s-
2009/02/15 10:07:39.312 00000FC4 HHCPN040I Instruction stepping off range 1048576-1056768
```

**Figure 139: S- command**

## 7.87 S? (Instruction stepping query)

### *7.87.1 Function*

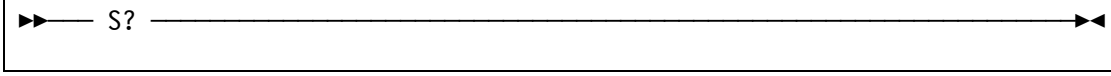
The S? command displays whether instruction stepping is on or off and the active range if any.

### *7.87.2 Syntax*

**Descriptive**

S?

**Diagram**




### *7.87.3 Parameter*

None.

### *7.87.4 Examples*

**Example 1:**

Query instruction stepping.

```
2009/02/15 10:08:05.112 00000FC4 s?
2009/02/15 10:08:05.112 00000FC4 HHCPN040I Instruction stepping on range 1048576-1056768
```

**Figure 140: S? command**

## 7.88 S{+/-} dev (Turn CCW stepping on or off)

### *7.88.1 Function*

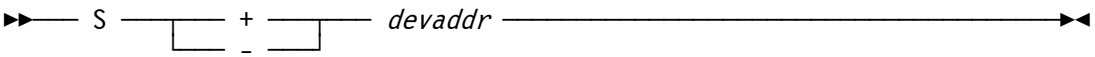
This command turns on (s+ dev) or turns off (s- dev) the CCW stepping function. After CCW stepping is turned on every execution of a CCW has to be confirmed by hitting ENTER on the Hercules console and produces detailed trace output.

### *7.88.2 Syntax*

**Descriptive**

S{+ | -} *devaddr*

**Diagram**




### *7.88.3 Parameter*

**devaddr** This is the address of the device for which CCW stepping will be turned on or off.

**+** The plus sign turns on the CCW stepping for the given device. The plus sign must immediately follow the S command (without an intervening blank).

**-** The minus sign turns off the CCW stepping for the given device. The minus sign must-immediately follow the S command (without an intervening blank).

### *7.88.4 Examples*

**Example 1:**

Turn on CCW stepping.

```
s+ 0148
HHCPN137I CCW stepping is now on for device 0148
0148:start i/o file[0] bufcur -1 cache[-1]
HHCCP048I 0148:CCW=02000000 60000018=>02000000 60000018 00000000 00000000 ....-...........
0148:HHCDA038I seeking to cyl 0 head 0
0148:read  trk   0 (asynchronous)
0148:0 rdtrk     0
0148:0 rdtrk[0] 0 cache miss
0148:0 rdtrk[0] 0 buf 00DA4FB0 len 56832
0148:trk[0] read_trkimg
0148:file[0] l2[0,0] trk[0] read_l2ent 0x710
```

```
0148:file[0] read_l2 0 active -1 -1 -1
0148:l2[0,0] cache[0] miss
0148:file[0] fd[5] read, off 0x710 len 2048
0148:file[0] cache[0] l2[0] read offset 0x710
0148:file[0] l2[0,0] trk[0] read_l2ent 0xf10 3057 3057
0148:file[0] fd[5] read, off 0xf10 len 3057
0148:0 rdtrk[0] 0 complete buf 00DA4FB0:0100000000
0148:uncompress comp 1 len 3057 maxlen 56832 trk 0
0148:newbuf malloc 00DBCED8 len 56832
0148:uncompress zlib newlen 6491 rc 0
0148:validating trk 0 len 6491 0000000000 0000000000000008
0148:read  trk   0 uncompressed len 6491
0148:HHCDA041I read count orientation is index
0148:HHCDA043I cyl 0 head 0 record 0 kl 0 dl 8 of 0
0148:HHCDA043I cyl 0 head 0 record 1 kl 4 dl 24 of 0
0148:HHCDA045I read data 24 bytes
HHCCP075I 0148:Stat=0C00 Count=0000 =>00080000 8000015A 06004A98 60000060 .......!...q-..-
HHCCP048I 0148:CCW=06004A98 60000060=>00000000 00000000 00000000 00000000 ................
0148:HHCDA041I read count orientation is data
0148:HHCDA043I cyl 0 head 0 record 2 kl 4 dl 144 of 0
0148:HHCDA045I read data 144 bytes
HHCCP075I 0148:Stat=0C00 Count=0000 =>07004AB8 40000006 31004ABE 40000005 .... ....... ...
HHCCP048I 0148:CCW=08004A98 00000000=>07004AB8 40000006 31004ABE 40000005 .... ....... ...
HHCCP048I 0148:CCW=07004AB8 40000006=>00000000 00000000 00000400 00000000 ................
0148:HHCDA038I seeking to cyl 0 head 0
HHCCP075I 0148:Stat=0C00 Count=0000
HHCCP048I 0148:CCW=31004ABE 40000005=>00000000 04000000 00000000 00000000 ................
0148:HHCDA041I read count orientation is index
0148:HHCDA043I cyl 0 head 0 record 0 kl 0 dl 8 of 0
HHCCP075I 0148:Stat=0C00 Count=0000
HHCCP048I 0148:CCW=08004AA0 00000000=>31004ABE 40000005 08004AA0 00000000 .... ...........
HHCCP048I 0148:CCW=31004ABE 40000005=>00000000 04000000 00000000 00000000 ................
0148:HHCDA041I read count orientation is count
0148:HHCDA043I cyl 0 head 0 record 1 kl 4 dl 24 of 0
HHCCP075I 0148:Stat=0C00 Count=0000
HHCCP048I 0148:CCW=08004AA0 00000000=>31004ABE 40000005 08004AA0 00000000 .... ...........
HHCCP048I 0148:CCW=31004ABE 40000005=>00000000 04000000 00000000 00000000 ................
0148:HHCDA041I read count orientation is count
0148:HHCDA043I cyl 0 head 0 record 2 kl 4 dl 144 of 0
HHCCP075I 0148:Stat=0C00 Count=0000
HHCCP048I 0148:CCW=08004AA0 00000000=>31004ABE 40000005 08004AA0 00000000 .... ...........
HHCCP048I 0148:CCW=31004ABE 40000005=>00000000 04000000 00000000 00000000 ................
0148:HHCDA041I read count orientation is count
0148:HHCDA043I cyl 0 head 0 record 3 kl 4 dl 80 of 0
HHCCP075I 0148:Stat=0C00 Count=0000
HHCCP048I 0148:CCW=08004AA0 00000000=>31004ABE 40000005 08004AA0 00000000 .... ...........
HHCCP048I 0148:CCW=31004ABE 40000005=>00000000 04000000 00000000 00000000 ................
.
.
.
```

**Figure 141: S+ dev command**

**Example 2:**

Turn off CCW stepping.

```
2007/06/30 12:40:00.328 00000FC4 s- 0148
2007/06/30 12:40:00.328 00000FC4 HHCPN137I CCW stepping is now off for device 0148
```

**Figure 142: S- dev command**

## 7.89 SAVECORE (Save a core image to a file)

### *7.89.1 Function*

The SAVECORE command allows you to save a a portion of real storage in binary form to a file. This command is used mainly for emulator debugging purposes. A certain core snapshot can be saved and then be restored at any later time to reproduce some tests with identical real storage values.

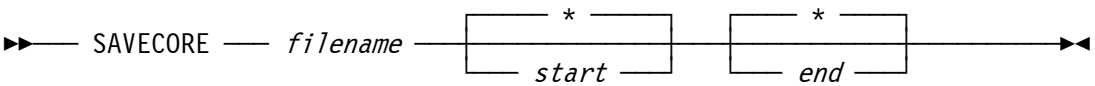
Please note that you have to stop all CPUs (see STOP/STOPALL commands) before saving the core image to a file.

### *7.89.2 Syntax*

**Descriptive**

```
SAVECORE filename [start | * ] [end | * ]
```

**Diagram**




### *7.89.3 Parameter*

**filename** This argument specifies the file name (and optionally the path) of the file where the core image will be written to.

**start | *** Start specifies the start address of the real storage to be saved to the file *filename*. The asterisk (“*”), the default, means the first byte of the first modified page as determined by the storage-key changed bit.

**end | *** End specifies the end address of the real storage to be saved to the file *filename*. The asterisk (“*”), the default, means the last byte of the last modified page as determined by the storage-key changed bit.

## *7.89.4 Examples*

**Example 1:**

Save the first 4096 bytes of real storage to file D:\core01.bin.

```
2007/06/30 09:50:40.156 0000008E stopall
2007/06/30 09:50:44.390 0000008E savecore d:\core01.bin 00000000 00001000
2007/06/30 09:50:44.390 0000008E HHCPN104I Saving locations 00000000-00001000 to d:\core01.bin
2007/06/30 09:50:44.406 0000008E HHCPN170I savecore command complete.
```

**Figure 143: SAVECORE command (save specific area)**

**Example 2:**

Save all real storage to file D:\core02.bin.

```
2007/06/30 09:51:10.156 0000008E stopall
2007/06/30 09:51:22.640 0000008E savecore d:\core02.bin * *
2007/06/30 09:51:22.640 0000008E HHCPN104I Saving locations 00000000-0FFFEFFF to d:\core02.bin
2007/06/30 09:51:29.500 0000008E HHCPN170I savecore command complete.
```

**Figure 144: SAVECORE command (save from begin to end)**

## 7.90 SCLPROOT (Set or display SCLP base directory)

### *7.90.1 Function*

The SCLPROOT command sets or displays the SCLP base directory. If a directory is given then SCLP disk I/O for the specified directory path is enabled. NONE disables SCLP disk I/O.

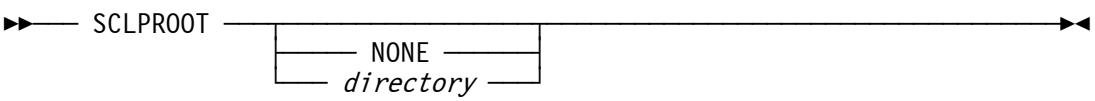
A subsequent list-directed IPL resets the path to the location of the .ins file, and a subsequent CCW-type IPL disables SCLP disk I/O. If no operand is specified, SCLPROOT displays the current setting.

### *7.90.2 Syntax*

**Descriptive**

```
SCLPROOT [NONE | directory]
```

**Diagram**




### *7.90.3 Parameter*

**NONE** Disables SCLP disk I/O.

**directory** Specifies the directory from which SCLP disk I/O is allowed. A subsequent IPL of an .ins file or a subsequent CCW-type IPL will override this.

### *7.90.4 Examples*

**Example 1:**

Display SCLP base directory.

```
2009/03/07 21:30:15.796 00000D80 sclproot
2009/03/07 21:30:15.796 00000D80 SCLP DISK I/O Disabled
```

**Figure 145: SCLPROOT command (display directory)**

**Example 2:**

Set SCLP base directory.

```
2009/03/07 21:30:27.109 00000D80 sclproot D:/ISO
```

**Figure 146: SCLPROOT command (set directory)**

## 7.91 SCP (Send system control program command)

### *7.91.1 Function*

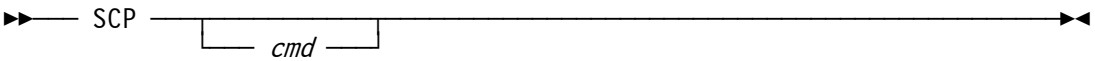
The SCP command sends a command to the system control program (i.e. the guest operating system running under Hercules) in any CMDTGT mode.

### *7.91.2 Syntax*

**Descriptive**

SCP [*cmd*]

**Diagram**




### *7.91.3 Parameter*

**cmd** This is the command to be sent to the system control program (the guest operating system).

### *7.91.4 Examples*

**Example 1:**

Send a command to the system control program running under Hercules when in HERC or PSCP command target mode.

```
2008/08/30 09:09:04.406 00000FC4 *   IGGN504A SPECIFY UNIT FOR CATALOG.MVS.MASTER ON MVSRES
2008/08/30 09:10:04.406 00000FC4 scp r 00,0148
2008/08/30 09:10:04.406 00000FC4 HHCCP041I SYSCONS interface active
2008/08/30 09:10:04.406 00000FC4  IEE600I REPLY TO 00 IS;0148
```

**Figure 147: SCP command**

## 7.92 SCRIPT (Run a sequence of commands contained in a file)

### *7.92.1 Function*

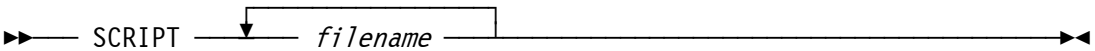
The SCRIPT command sequentially executes the commands contained within the file *filename*. The script file itself may also contain script commands but the system ensures that no more than 10 levels of scripts are invoked at any one time to avoid recursion loops.

### *7.92.2 Syntax*

**Descriptive**

SCRIPT *filename* [*filename* ... ]

**Diagram**




### *7.92.3 Parameter*

**filename** The name (and optionally the path) of the script file to be executed.

### *7.92.4 Examples*

**Example 1:**

Execute script “HERCULES.RC”.

```
2007/06/30 13:10:53.125 00000FC4 script hercules.rc
2007/06/30 13:10:53.125 00000FC4 sh telnet localhost 3270
2007/06/30 13:10:53.140 00000FC4 ?
2007/06/30 13:10:53.140 00000FC4 HHCPN140I Valid panel commands are...
2007/06/30 13:10:53.140 00000FC4
2007/06/30 13:10:53.140 00000FC4   Command      Description...
2007/06/30 13:10:53.140 00000FC4   -------      ----------------------------------------------
2007/06/30 13:10:53.140 00000FC4   ?            list all commands
.
.
.
```

**Figure 148: SCRIPT command**

## 7.93 SCSIMOUNT (Automatic SCSI tape mounts)

### *7.93.1 Function*

The SCSIMOUNT command displays or modifies the automatic SCSI tape mount option. When entered without any operands it displays the current value and any pending tape mount requests. Entering a value between 1 and 99 enables the option and specifies how often (in seconds) to query SCSI tape drives to automatically detect when a tape has been mounted. When a tape is mounted an unsolicited device attention interrupt will be presented to the guest operating system.

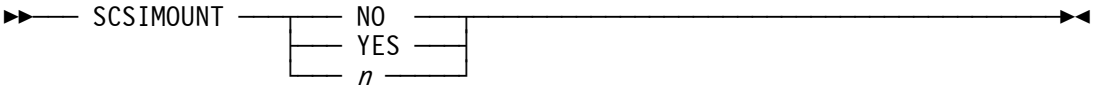
Notes: Enabling this option may negatively impact Hercules performance depending on how the host operating system (Windows, Linux, etc.) processes SCSI attached tape drive status queries. SCSI-MOUNT is a synonym for the AUTO_SCSI_MOUNT command.

### *7.93.2 Syntax*

**Descriptive**

```
SCSIMOUNT [NO | YES | n]
```

**Diagram**




### *7.93.3 Parameter*

**NO** No indicates that the SCSIMOUNT option is disabled forcing all SCSI tape mounts to be done manually via an appropriate DEVINIT command.

**YES** Yes enables the option and causes periodic queries of the SCSI tape drive in a five second interval to automatically detect when a tape is mounted. YES is equivalent with “SCSIMOUNT 5”.

**n** A value from 1 to 99 seconds inclusive enables the option and causes periodic queries of the SCSI tape drive to automatically detect when a tape is mounted.

### *7.93.4 Examples*

**Example 1:**

Enable the SCSIMOUNT option.

```
2007/06/30 10:57:06.296 00000FC4 scsimount 30
2007/06/30 10:57:06.296 00000FC4 SCSI auto-mount queries = every 30 seconds (when needed)
```

**Figure 149: SCSIMOUNT command**

## 7.94 SF+ (Create a new shadow file)

### *7.94.1 Function*

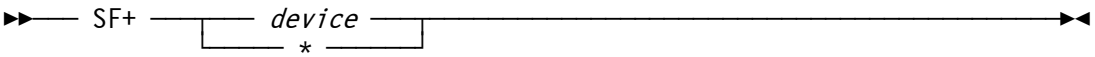
With the SF+ command a new shadow file for a certain dasd device (or for all dasd devices that have shadow files) is created.

### *7.94.2 Syntax*

**Descriptive**

SF+ {*device* | *}

**Diagram**




### *7.94.3 Parameter*

**device** This specifies the device for which a new shadow file will be created.

***** With an asterisk it is possible to create a (new) shadow file for all dasd devices that have a shadow file defined in their configuration file definition statements.

### *7.94.4 Examples*

**Example 1:**

Create a new shadow file for device 0148.

```
00000FC4 sf+ 0148
00000FC4 HHCCD162I 0148 file[1] D:/MVS/DASD/MVSRES_1.148 added
00000FC4 HHCCD210I           size free  nbr st   reads  writes l2reads    hits switches
00000FC4 HHCCD212I ------------------------------------------------------------------
00000FC4 HHCCD213I [*]  324636047   6%   13          0       0       0       0        0
00000FC4 HHCCD215I D:/MVS/DASD/MVSRES.CCKD
00000FC4 HHCCD216I [0]  324634239   6%   13 rd       0       0       0
00000FC4 HHCCD218I [1]       1808   0%    0 rw       0       0       0
```

**Figure 150: SF+ command**

## 7.95 SF- (Remove a shadow file)

### *7.95.1 Function*

The SF- command removes a shadow file. Depending on the MERGE / NOMERGE / FORCE parameter the changes are incorporated in the base file or discarded.

If MERGE is specified or defaulted then the contents of the current file is merged into the previous file, the current file is removed and the previous file becomes the current file.

If NOMERGE is specified then the contents of the current file is discarded and the previous file becomes the current file. However if the previous file is read-only then a new shadow file is created ("re-added") and that becomes the current file.

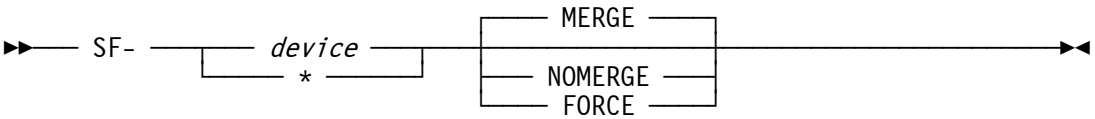
The FORCE option is required when doing a merge to a base file and the base file is read-only because the 'ro' option was specified on the device configuration statement.

### *7.95.2 Syntax*

**Descriptive**

SF- {*device* | *} [MERGE | NOMERGE | FORCE]

**Diagram**




### *7.95.3 Parameter*

**device** This specifies the device for which a shadow file should be deleted.

***** With an asterisk it is possible to remove a shadow file for all dasd devices that have shadow files.

**MERGE** The MERGE parameter (which is the default) specifies that all changes or updates that have been made to the shadow file will be commited.

**NOMERGE** The NOMERGE parameter specifies that all changes or updates that have been made to the shadow file will be discarded instead of being committed.

**FORCE** The FORCE parameter is used when doing a merge to a base file that is read-only because the 'ro' option was specified on the device configuration statement.

## *7.95.4 Examples*

**Example 1:**

Remove a shadow file with backwards merge (i.e. *commit* all of the changes/updates).

```
00000FC4 sf- 0148
00000FC4 HHCCD181I 0148 shadow file [1] successfully merged
00000FC4 HHCCD210I           size free  nbr st   reads  writes l2reads    hits switches
00000FC4 HHCCD211I                                                readaheads   misses
00000FC4 HHCCD212I ------------------------------------------------------------------------
00000FC4 HHCCD213I [*]  324634239   6%   22       4296    1134      45    4395     7267
00000FC4 HHCCD214I                                                      1426      126
00000FC4 HHCCD215I D:/MVS/DASD/MVSRES.CCKD
00000FC4 HHCCD216I [0]  324634239   6%   22 rw    4053       0      45
00000FC4 HHCCD217I D:/MVS/DASD/MVSRES_*.148
```

**Figure 151: SF- command**

**Example 2:**

Remove a shadow file without backwards merge (i.e. *discard* all of the changes/updates).

```
00000FC4 sf- 0148 nomerge
00000FC4 HHCCD181I 0148 shadow file [1] successfully removed
00000FC4 HHCCD210I           size free  nbr st   reads  writes l2reads    hits switches
00000FC4 HHCCD211I                                                readaheads   misses
00000FC4 HHCCD212I ------------------------------------------------------------------------
00000FC4 HHCCD213I [*]  324634239   6%   22       4329    1118      45    4376     7268
00000FC4 HHCCD214I                                                      1440      140
00000FC4 HHCCD215I D:/MVS/DASD/MVSRES.CCKD
00000FC4 HHCCD216I [0]  324634239   6%   22 rw    4061       0      45
00000FC4 HHCCD217I D:/MVS/DASD/MVSRES_*.148
```

**Figure 152: SF- NOMERGE command**

## 7.96 SFC (Compress a shadow file)

### *7.96.1 Function*

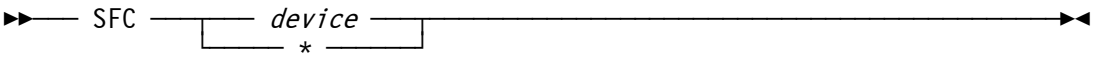
The SFC command compresses a shadow file.

### *7.96.2 Syntax*

**Descriptive**

SFC {*device* | *}

**Diagram**




### *7.96.3 Parameter*

**device** This specifies the device for which the shadow file will be compressed.

***** With an asterisk it is possible to compress the shadow files for all dasd devices that have a shadow file.

### *7.96.4 Examples*

**Example 1:**

Compress the shadow file for dasd device 0148.

```
00000FC4 sfc 0148
00000FC4 cckdcomp: completed: moves 1105  moved 23817149  freed 299782
00000FC4 HHCCD210I           size free  nbr st   reads  writes l2reads    hits switches
00000FC4 HHCCD211I                                                readaheads   misses
00000FC4 HHCCD212I ------------------------------------------------------------------------
00000FC4 HHCCD213I [*]  348465484   6%   22       4329    1118      45    4376     7268
00000FC4 HHCCD214I                                                     1440      140
00000FC4 HHCCD215I D:/MVS/DASD/MVSRES.CCKD
00000FC4 HHCCD216I [0]  324634239   6%   22 rd    4061       0      45
00000FC4 HHCCD217I D:/MVS/DASD/MVSRES_*.148
00000FC4 HHCCD218I [1]   23831245   0%    0 rw     268    1118       0
```

**Figure 153: SFC command**

## 7.97 SFD (Display shadow file statistics)

### *7.97.1 Function*

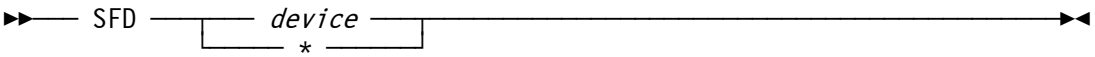
The SFD command displays statistical information about the specified shadow file(s).

### *7.97.2 Syntax*

**Descriptive**

SFD {*device* | *}

**Diagram**




### *7.97.3 Parameter*

**device** This specifies the device for which shadow file statistics will be written to the log file.

***** With an asterisk it is possible to create statistical output for all dasd devices that have shadow files.

### *7.97.4 Examples*

**Example 1:**

Display statistical information for the shadow file of dasd device 0148.

```
00000FC4 sfd 0148
00000FC4 HHCCD210I           size free  nbr st   reads  writes l2reads    hits switches
00000FC4 HHCCD211I                                                readaheads   misses
00000FC4 HHCCD212I ------------------------------------------------------------------
00000FC4 HHCCD213I [*]  348613498   6%   30       4329    1112      45    4376     7268
00000FC4 HHCCD214I                                                      1440      140
00000FC4 HHCCD215I D:/MVS/DASD/MVSRES.CCKD
00000FC4 HHCCD216I [0]  324634239   6%   22 rd    4061       0      45
00000FC4 HHCCD217I D:/MVS/DASD/MVSRES_*.148
00000FC4 HHCCD218I [1]   23979259   0%    8 rw     268    1112       0
```

**Figure 154: SFD command**

## 7.98 SFK (Perform a chkdsk on the active shadow file)

### *7.98.1 Function*

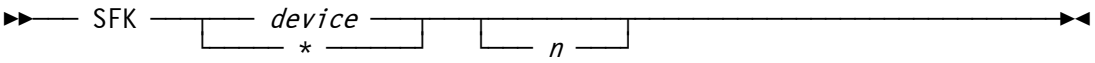
The SFK command performs a chkdsk on the active shadow file(s).

### *7.98.2 Syntax*

**Descriptive**

SFK {*device* | *} [*n*]

**Diagram**




### *7.98.3 Parameter*

**device** This specifies the device for which active shadow file a chkdsk is to be performed.

***** With an asterisk it is possible to perform a chkdsk for the active shadow files of all dasd devices.

**n** The optional check level to be performed (default is 2):

| | |
|---|---|
| -1 | devhdr, cdevhdr, I1 table |
| 0 | devhdr, cdevhdr, I1 table, I2 tables |
| 1 | devhdr, cdevhdr, I1 table, I2 tables, free spaces |
| 2 | devhdr, cdevhdr, I1 table, I2 tables, free spaces, trkhdrs |
| 3 | devhdr, cdevhdr, I1 table, I2 tables, free spaces, trkimgs |
| 4 | devhdr, cdevhdr, build everything else from recovery |

Use check level '4' only after making a backup and be prepared to wait a long time!

## *7.98.4 Examples*

**Example 1:**

Perform chkdsk level '2' for the shadow file of dasd device 0AE0.

```
00000AE0 sfk 0148
00000AE0 HHCCD210I           size free  nbr st   reads  writes l2reads    hits switches
00000AE0 HHCCD211I                                                readaheads   misses
00000AE0 HHCCD212I ------------------------------------------------------------------------
00000AE0 HHCCD213I [*]  340102838   0%    0       8823     836     142   27681    33892
00000AE0 HHCCD214I                                                     2614     1191
00000AE0 HHCCD215I D:/MVS/DASD/MVSRES.CCKD
00000AE0 HHCCD216I [0]  340048223   0%    0 rd    8570       0     137
00000AE0 HHCCD217I D:/MVS/DASD/MVSRES*.148
00000AE0 HHCCD218I [1]      54615   0%    0 rw     253     836       5
```

**Figure 155: SFK command (chkdsk level 2)**

**Example 2:**

Perform chkdsk level '4' for the shadow file of dasd device 0AE0.

```
00000AE0 sfk 0AE0
00000AE0 sfk 0148 4
00000AE0 HHCCU301I 0AE0 file[1]: trk[26309] recovered offset 0x2f10 len 57
00000AE0 HHCCU301I 0AE0 file[1]: trk[34077] recovered offset 0x2f49 len 125
00000AE0 HHCCU301I 0AE0 file[1]: trk[34047] recovered offset 0x2fc6 len 208
00000AE0 HHCCU301I 0AE0 file[1]: trk[34050] recovered offset 0x3096 len 104
00000AE0 HHCCU301I 0AE0 file[1]: trk[7] recovered offset 0x30fe len 924
00000AE0 HHCCU301I 0AE0 file[1]: trk[34079] recovered offset 0x349a len 210
00000AE0 HHCCU301I 0AE0 file[1]: trk[34048] recovered offset 0x356c len 769
00000AE0 HHCCU301I 0AE0 file[1]: trk[34081] recovered offset 0x386d len 142
00000AE0 HHCCU301I 0AE0 file[1]: trk[34080] recovered offset 0x38fb len 765
00000AE0 HHCCU301I 0AE0 file[1]: trk[34082] recovered offset 0x3bf8 len 105
00000AE0 HHCCU301I 0AE0 file[1]: trk[34078] recovered offset 0x3c61 len 349
00000AE0 HHCCU301I 0AE0 file[1]: trk[34045] recovered offset 0x3dbe len 125
00000AE0 HHCCU301I 0AE0 file[1]: trk[34046] recovered offset 0x3e3b len 349
00000AE0 HHCCU301I 0AE0 file[1]: trk[34075] recovered offset 0x3f98 len 279
00000AE0 HHCCU301I 0AE0 file[1]: trk[34084] recovered offset 0x40af len 341
00000AE0 HHCCU301I 0AE0 file[1]: trk[34085] recovered offset 0x4204 len 341
00000AE0 HHCCU301I 0AE0 file[1]: trk[34049] recovered offset 0x4359 len 143
00000AE0 HHCCU301I 0AE0 file[1]: trk[34083] recovered offset 0x43e8 len 631
00000AE0 HHCCU301I 0AE0 file[1]: trk[34086] recovered offset 0x465f len 340
00000AE0 HHCCU301I 0AE0 file[1]: trk[34088] recovered offset 0x47b3 len 339
00000AE0 HHCCU301I 0AE0 file[1]: trk[34103] recovered offset 0x4906 len 292
00000AE0 HHCCU301I 0AE0 file[1]: trk[34089] recovered offset 0x4a2a len 339
00000AE0 HHCCU301I 0AE0 file[1]: trk[34090] recovered offset 0x4b7d len 340
00000AE0 HHCCU301I 0AE0 file[1]: trk[31995] recovered offset 0x4cd1 len 3870
00000AE0 HHCCU301I 0AE0 file[1]: trk[34087] recovered offset 0x5bef len 340
00000AE0 HHCCU301I 0AE0 file[1]: trk[34091] recovered offset 0x5d43 len 340
00000AE0 HHCCU301I 0AE0 file[1]: trk[34092] recovered offset 0x5e97 len 340
00000AE0 HHCCU301I 0AE0 file[1]: trk[31999] recovered offset 0x5feb len 7932
```

```
00000AE0 HHCCU301I 0AE0 file[1]: trk[4] recovered offset 0x7ee7 len 1320
00000AE0 HHCCU301I 0AE0 file[1]: trk[5] recovered offset 0x840f len 1526
00000AE0 HHCCU301I 0AE0 file[1]: trk[3] recovered offset 0x8a05 len 1445
00000AE0 HHCCU301I 0AE0 file[1]: trk[34093] recovered offset 0x8faa len 340
00000AE0 HHCCU301I 0AE0 file[1]: trk[34094] recovered offset 0x90fe len 340
00000AE0 HHCCU301I 0AE0 file[1]: trk[34095] recovered offset 0x9252 len 338
00000AE0 HHCCU301I 0AE0 file[1]: trk[34096] recovered offset 0x93a4 len 339
00000AE0 HHCCU301I 0AE0 file[1]: trk[34097] recovered offset 0x94f7 len 340
00000AE0 HHCCU301I 0AE0 file[1]: trk[34098] recovered offset 0x964b len 340
.
several lines not displayed
.
00000AE0 HHCCU301I 0AE0 file[1]: trk[34060] recovered offset 0xa9fd len 339
00000AE0 HHCCU301I 0AE0 file[1]: trk[34061] recovered offset 0xab50 len 340
00000AE0 HHCCU301I 0AE0 file[1]: trk[34062] recovered offset 0xaca4 len 340
00000AE0 HHCCU301I 0AE0 file[1]: trk[34063] recovered offset 0xadf8 len 340
00000AE0 HHCCU301I 0AE0 file[1]: trk[34064] recovered offset 0xaf4c len 340
00000AE0 HHCCU301I 0AE0 file[1]: trk[34065] recovered offset 0xb0a0 len 338
00000AE0 HHCCU301I 0AE0 file[1]: trk[34066] recovered offset 0xb1f2 len 338
00000AE0 HHCCU301I 0AE0 file[1]: trk[34067] recovered offset 0xb344 len 340
00000AE0 HHCCU301I 0AE0 file[1]: trk[34068] recovered offset 0xb498 len 340
00000AE0 HHCCU301I 0AE0 file[1]: trk[34069] recovered offset 0xb5ec len 340
00000AE0 HHCCU301I 0AE0 file[1]: trk[34070] recovered offset 0xb740 len 340
00000AE0 HHCCU301I 0AE0 file[1]: trk[34071] recovered offset 0xb894 len 295
00000AE0 HHCCU301I 0AE0 file[1]: trk[1] recovered offset 0xb9bb len 1570
00000AE0 HHCCU301I 0AE0 file[1]: trk[34107] recovered offset 0xbfdd len 277
00000AE0 HHCCU301I 0AE0 file[1]: trk[6] recovered offset 0xc0f2 len 796
00000AE0 HHCCU301I 0AE0 file[1]: trk[34072] recovered offset 0xc40e len 349
00000AE0 HHCCU301I 0AE0 file[1]: trk[34073] recovered offset 0xc56b len 338
00000AE0 HHCCU301I 0AE0 file[1]: trk[34074] recovered offset 0xc6bd len 340
00000AE0 HHCCU301I 0AE0 file[1]: trk[34104] recovered offset 0xc811 len 345
00000AE0 HHCCU301I 0AE0 file[1]: trk[34105] recovered offset 0xc96a len 338
00000AE0 HHCCU301I 0AE0 file[1]: trk[34106] recovered offset 0xcabc len 338
00000AE0 HHCCU301I 0AE0 file[1]: trk[8] recovered offset 0xcc0e len 665
00000AE0 HHCCU301I 0AE0 file[1]: trk[2] recovered offset 0xcea7 len 1712
00000AE0 HHCCU300I 0AE0 file[1]: 73 trk images recovered
00000AE0 HHCCU104I 0AE0 file[1]: free space rebuilt
00000AE0 HHCCD210I           size free  nbr st   reads  writes l2reads    hits switches
00000AE0 HHCCD211I                                                 readaheads   misses
00000AE0 HHCCD212I ----------------------------------------------------------------------
00000AE0 HHCCD213I [*]  340102838   0%    0       8825     836     143   27681    33894
00000AE0 HHCCD214I                                                      2614     1191
00000AE0 HHCCD215I D:/MVS/DASD/MVSRES.CCKD
00000AE0 HHCCD216I [0]  340048223   0%    0 rd    8572       0     138
00000AE0 HHCCD217I D:/MVS/DASD/MVSRES*.CCKD
00000AE0 HHCCD218I [1]      54615   0%    0 rw     253     836       5
```

**Figure 156: SFK command (chkdsk level 4)**

## 7.99 SH (Shell command)

### *7.99.1 Function*

The SH command passes the given command and the parameters (if any) as-is to the shell for processing. The results of the passed command are displayed on the Hercules console.

The special 'STARTGUI' command must be used if the command being started either directly or indirectly starts a Windows graphical user interface (non command-line) program (e.g. Notepad). If STARTGUI is not used in such cases then Hercules will hang until the graphical user interface program is closed. Note that starting a batch file (command line program) that itself starts a graphical user interface program still requires using STARTGUI.

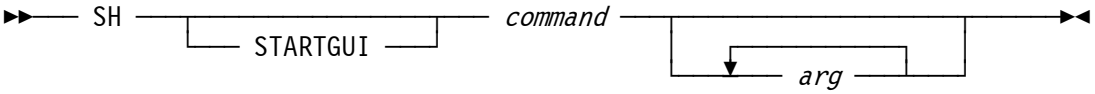
If "foo.bat" contains "start notepad.exe" then issuing "sh foo.bat" will hang Hercules until Notepad is being closed. The same applies if "sh start foo.bat" is used. The correct way is using 'STARTGUI' for invoking foo.bat is "sh startgui foo.bat".

### *7.99.2 Syntax*

**Descriptive**

SH [STARTGUI] *command* [*arg* [*arg* ... ]]

**Diagram**




### *7.99.3 Parameter*

**STARTGUI** STARTGUI must be used if the command passed to the shell either directly or indirectly starts a Windows graphical user interface (non command-line) program.

**command** The command that is to be passed to the shell.

***arg*** These are the parameters passed to the command.

## *7.99.4 Examples*

**Example 1:**

Display the current working directory through the shell command 'DIR'.

```
2007/06/30 16:11:56.140 sh dir
2007/06/30 16:11:56.156 Command started; process-id=00000550
2007/06/30 16:11:56.250 00000550  Volume in drive D is DATA
2007/06/30 16:11:56.250 00000550  Volume Serial Number is 44C3-18D9
2007/06/30 16:11:56.250 00000550
2007/06/30 16:11:56.250 00000550  Directory of D:\MVS
2007/06/30 16:11:56.250 00000550
2007/06/30 16:11:56.250 00000550 07.08.2006  17:59    <DIR>          .
2007/06/30 16:11:56.250 00000550 07.08.2006  17:59    <DIR>          ..
2007/06/30 16:11:56.250 00000550 07.08.2006  17:59    <DIR>          BATCH
2007/06/30 16:11:56.265 00000550 07.08.2006  17:59    <DIR>          CONF
2007/06/30 16:11:56.265 00000550 07.08.2006  17:59    <DIR>          DASD
2007/06/30 16:11:56.265 00000550 07.08.2006  17:59    <DIR>          JCL
2007/06/30 16:11:56.265 00000550 25.08.2006  16:11    <DIR>          LOG
2007/06/30 16:11:56.265 00000550 07.08.2006  17:59    <DIR>          PCH
2007/06/30 16:11:56.265 00000550 07.08.2006  17:59    <DIR>          PRT
2007/06/30 16:11:56.265 00000550 07.08.2006  17:59    <DIR>          SHADOW
2007/06/30 16:11:56.265 00000550 07.08.2006  17:59    <DIR>          TAPE
2007/06/30 16:11:56.265 00000550 07.08.2006  17:59    <DIR>          TN3270
2007/06/30 16:11:56.265 00000550 12.08.1994  12:23             8'480 SLEEP.EXE
2007/06/30 16:11:56.265 00000550 24.10.2005  07:57               633 MVS38J_GUI.BAT
2007/06/30 16:11:56.265 00000550 24.10.2005  08:23               632 MVS38J_NATIVE.BAT
2007/06/30 16:11:56.265 00000550 24.10.2005  08:18             1'053 MVS38J_TERMINAL.BAT
2007/06/30 16:11:56.265 00000550                4 File(s)         10'798 bytes
2007/06/30 16:11:56.265 00000550               12 Dir(s)  25'160'966'144 bytes free
2007/06/30 16:11:56.265 Command ended; rc=0
```

**Figure 157: SH command**

# 7.100 SHRD (SHRD command)

## 7.100.1 Function

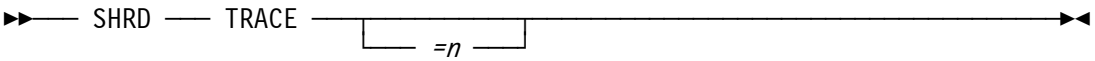
The SHRD command has two functions. Depending on the given arguments it initializes the Shared Device Server trace table and starts the trace or it prints the contents of the current trace table.

## 7.100.2 Syntax

**Descriptive**

```
SHRD TRACE[=n]
```

**Diagram**




## 7.100.3 Parameter

**TRACE** TRACE, when specified without a decimal number, prints the current contents of the trace table.

**n** This specifies the trace table size (decimal). If the TRACE argument is given with a number then a trace table with a size of *n* elements initialized.

## 7.100.4 Examples

**Example 1:**

Initialize the Shared Device Server trace table with a size of 2500 trace entries.

```
2007/08/25 17:36:48.153 00000190 shrd trace=2500
```

**Figure 158: SHRD command (initialize trace table)**

**Example 2:**

Print the Shared Device Server trace table.

```
2007/08/25 17:37:57.091 00000190 shrd trace
2007/08/25 17:37:58.091 00000190 1188056256.238492 0148:select rc 1
2007/08/25 17:37:58.106 00000190 1188056256.238502 0148:select isset 1048 id=1
2007/08/25 17:37:58.106 00000190 1188056256.238509 0148:select ready 1048 id=1
2007/08/25 17:37:58.106 00000190 1188056256.238519 0000:recvData    e3 00 148 1 0
2007/08/25 17:37:58.106 00000190 1188056256.238524 0148:server_request [0] e3 00 148 1 0
2007/08/25 17:37:58.106 00000190 1188056256.238530 0148:server_request inactive id=1
2007/08/25 17:37:58.106 00000190 1188056256.238534 0148:server_send 00 00 148 1 0
2007/08/25 17:37:58.106 00000190 1188056256.238560 0148:select   set 1048 id=1
2007/08/25 17:37:58.106 00000190 1188056256.238988 0148:select rc 1
2007/08/25 17:37:58.106 00000190 1188056256.238999 0148:select isset 1048 id=1
2007/08/25 17:37:58.106 00000190 1188056256.239006 0148:select ready 1048 id=1
2007/08/25 17:37:58.106 00000190 1188056256.239020 0000:recvData    e2 00 148 1 0
2007/08/25 17:37:58.106 00000190 1188056256.239028 0148:server_request [0] e2 00 148 1 0
2007/08/25 17:37:58.106 00000190 1188056256.239035 0148:server_request active id=1
2007/08/25 17:37:58.106 00000190 1188056256.239041 0148:server_send 00 00 148 1 0
2007/08/25 17:37:58.106 00000190 1188056256.239075 0148:select   set 1048 id=1
2007/08/25 17:37:58.106 00000190 1188056256.239243 0148:select rc 1
2007/08/25 17:37:58.106 00000190 1188056256.239252 0148:select isset 1048 id=1
2007/08/25 17:37:58.106 00000190 1188056256.239260 0148:select ready 1048 id=1
2007/08/25 17:37:58.106 00000190 1188056256.239270 0000:recvData    e8 00 148 1 4
2007/08/25 17:37:58.106 00000190 1188056256.239278 0148:server_request [0] e8 00 148 1 4
.
several lines not displyed
.
2007/08/25 17:37:58.356 00000190 1188056256.591571 0148:server_request inactive id=1
2007/08/25 17:37:58.356 00000190 1188056256.591579 0148:server_send 00 00 148 1 0
2007/08/25 17:37:58.356 00000190 1188056256.591630 0148:select   set 1048 id=1
2007/08/25 17:37:58.356 00000190 1188056259.840521 0340:select rc 0
2007/08/25 17:37:58.356 00000190 1188056259.840537 0340:select   set 1020 id=1
2007/08/25 17:37:58.356 00000190 1188056259.840538 0341:select rc 0
2007/08/25 17:37:58.356 00000190 1188056259.840547 0341:select   set 992 id=1
2007/08/25 17:37:58.356 00000190 1188056259.840563 0342:select rc 0
2007/08/25 17:37:58.356 00000190 1188056259.840569 0343:select rc 0
2007/08/25 17:37:58.356 00000190 1188056259.840572 0342:select   set 964 id=1
2007/08/25 17:37:58.356 00000190 1188056259.840579 0343:select   set 936 id=1
2007/08/25 17:37:58.356 00000190 1188056266.591236 0148:select rc 0
2007/08/25 17:37:58.356 00000190 1188056266.591253 0148:select   set 1048 id=1
2007/08/25 17:37:58.356 00000190 1188056269.841130 0340:select rc 0
2007/08/25 17:37:58.356 00000190 1188056269.841149 0340:select   set 1020 id=1
2007/08/25 17:37:58.356 00000190 1188056269.841177 0342:select rc 0
2007/08/25 17:37:58.356 00000190 1188056269.841186 0342:select   set 964 id=1
2007/08/25 17:37:58.356 00000190 1188056269.841187 0341:select rc 0
2007/08/25 17:37:58.356 00000190 1188056269.841197 0341:select   set 992 id=1
2007/08/25 17:37:58.356 00000190 1188056269.841220 0343:select rc 0
2007/08/25 17:37:58.356 00000190 1188056269.841229 0343:select   set 936 id=1
2007/08/25 17:37:58.356 00000190 1188056276.590940 0148:select rc 0
2007/08/25 17:37:58.356 00000190 1188056276.590957 0148:select   set 1048 id=1
```

**Figure 159: SHRD command (print trace table)**

## 7.101 SIZEOF (Display size of structures)

### *7.101.1 Function*

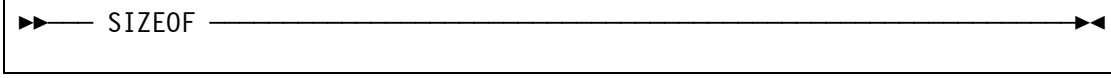
The SIZEOF command shows the size of various Hercules internal structures.

### *7.101.2 Syntax*

**Descriptive**

```
SIZEOF
```

**Diagram**




### *7.101.3 Parameter*

None.

### *7.101.4 Examples*

**Example 1:**

Display the size of structures.

```
2009/02/07 18:41:40.191 00000830 sizeof
2009/02/07 18:41:40.191 00000830 HHCPN161I (void *) .........      4
2009/02/07 18:41:40.191 00000830 HHCPN161I (unsigned int) ....      4
2009/02/07 18:41:40.191 00000830 HHCPN161I (long) ............      4
2009/02/07 18:41:40.191 00000830 HHCPN161I (long long) .......      8
2009/02/07 18:41:40.191 00000830 HHCPN161I (size_t) .........      4
2009/02/07 18:41:40.191 00000830 HHCPN161I (off_t) ...........      8
2009/02/07 18:41:40.191 00000830 HHCPN161I SYSBLK ............  42848
2009/02/07 18:41:40.191 00000830 HHCPN161I REGS ..............  41624
2009/02/07 18:41:40.191 00000830 HHCPN161I REGS (copy len) ...   1480
2009/02/07 18:41:40.191 00000830 HHCPN161I PSW ................     40
2009/02/07 18:41:40.191 00000830 HHCPN161I DEVBLK ............   1816
2009/02/07 18:41:40.191 00000830 HHCPN161I TLB entry .........     36
2009/02/07 18:41:40.191 00000830 HHCPN161I TLB table .........  36864
2009/02/07 18:41:40.191 00000830 HHCPN161I FILENAME_MAX ......    260
2009/02/07 18:41:40.191 00000830 HHCPN161I PATH_MAX ..........    260
```

**Figure 160: SIZEOF command**

## 7.102 SSD (Signal shutdown)

### *7.102.1 Function*

The SSD (signal shutdown) command signals an imminent hypervisor shutdown to the guest. Guests who support this are supposed to perform a shutdown upon receiving this request.

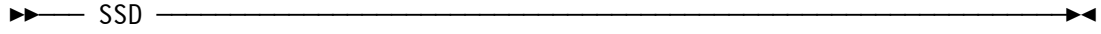
An implicit ssd command is given on a hercules QUIT command if the guest supports ssd. In that case the Hercules shutdown will be delayed until the guest has shutdown or a second quit command is given.

### *7.102.2 Syntax*

**Descriptive**

```
SSD
```

**Diagram**




### *7.102.3 Parameter*

None.

### *7.102.4 Examples*

**Example 1:**

Signal an imminent hypervisor shutdown to the guest operating system. In this example the guest operating system does not support this function.

```
2007/08/25 18:18:40.294 000004B8 ssd
2007/08/25 18:18:40.294 000004B8 HHCCP081E SCP not receiving quiesce signals
```

**Figure 161: SSD command**

## 7.103 START (Start CPU or printer device)

### 7.103.1 Function

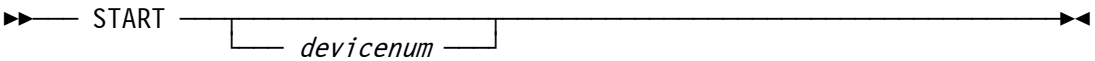
The START command by itself (without argument) simply starts a stopped CPU, whereas START *devicenum* presses the (virtual) start button on printer device *devicenum*.

### 7.103.2 Syntax

**Descriptive**

START [*devicenum*]

**Diagram**




### 7.103.3 Parameter

**devicenum** The device number of the printer which should be started.

### 7.103.4 Examples

**Example 1:**

Start the printer device 000E.

```
2007/06/30 10:07:43.296 00000FC4 start 000e
2007/06/30 10:07:43.296 00000FC4 HHCPN018I Printer 000E starter
```

**Figure 162: START command (start printer)**

**Example 2:**

Start CPU 1.

```
2007/06/30 10:09:23.148 00000FC4 cpu 1
2007/06/30 10:09:23.148 00000FC4 start
```

**Figure 163: START command (start CPU)**

## 7.104 STARTALL (Start all CPUs)

### *7.104.1 Function*

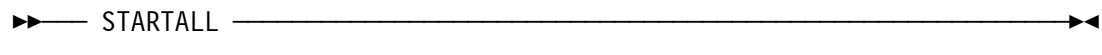
The STARTALL command starts all CPUs.

### *7.104.2 Syntax*

**Descriptive**

```
STARTALL
```

**Diagram**




### *7.104.3 Parameter*

None.

### *7.104.4 Examples*

**Example 1:**

Start all CPU's.

```
00000FC4 startall
00000FC4 HHCCP014I CPU0001: Operation exception CODE=0001 ILC=2
00000FC4 CPU0001:  PSW=00000000 00000000 INST=000A         ?????
00000FC4 CPU0001: GR00=00000000  GR01=00000000  GR02=00000000  GR03=00000000
00000FC4 CPU0001: GR04=00000000  GR05=00000000  GR06=00000000  GR07=00000000
00000FC4 CPU0001: GR08=00000000  GR09=00000000  GR10=00000000  GR11=00000000
00000FC4 CPU0001: GR12=00000000  GR13=00000000  GR14=00000000  GR15=00000000
00000FC4 HHCCP014I CPU0001: Addressing exception CODE=0005 ILC=6
00000FC4 CPU0001:  PSW=00080000 80000A1A INST=D50106401000 CLC   1600(2,0),0(1)
00000FC4 CPU0001:  R:00000640:K:06=B22C0005 A774FFFC 50405000 184518F5  ....x...& &....5
00000FC4 CPU0001:  R:7FFFFFFE: Translation exception 0005
00000FC4 CPU0001: GR00=00000000  GR01=FFFFFFFE  GR02=00000000  GR03=00000000
00000FC4 CPU0001: GR04=00000000  GR05=00000000  GR06=00000000  GR07=00000000
00000FC4 CPU0001: GR08=00000000  GR09=00000000  GR10=00000000  GR11=00000000
00000FC4 CPU0001: GR12=00000000  GR13=00000000  GR14=00000000  GR15=00000000
00000FC4 HHCCP011I CPU0001: Disabled wait state
00000FC4           PSW=000A0000 00000019
```

**Figure 164: STARTALL command**

## 7.105 STOP (Stop CPU or printer device)

### *7.105.1 Function*

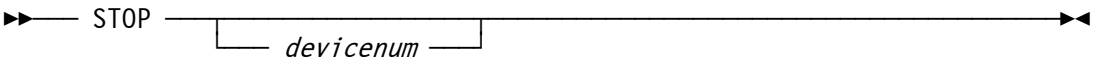
The STOP command by itself (without argument) simply stops a started CPU, whereas STOP *devicenum* presses the (virtual) stop button on printer device *devicenum*

### *7.105.2 Syntax*

**Descriptive**

STOP [*devicenum*]

**Diagram**




### *7.105.3 Parameter*

**devicenum** The device number of the printer which has to be stopped.

### *7.105.4 Examples*

**Example 1:**

Stop the printer device 000E.

```
2007/06/30 10:07:39.312 00000FC4 stop 000e
2007/06/30 10:07:39.312 00000FC4 HHCPN025I Printer 000E stopped
```

**Figure 165: STOP command (stop printer)**

**Example 2:**

Stop CPU 1.

```
2007/06/30 10:09:23.148 00000FC4 cpu 1
2007/06/30 10:09:23.148 00000FC4 stop
```

**Figure 166: STOP command (stop CPU)**

## 7.106 STOPALL (Stop all CPUs)

### *7.106.1 Function*

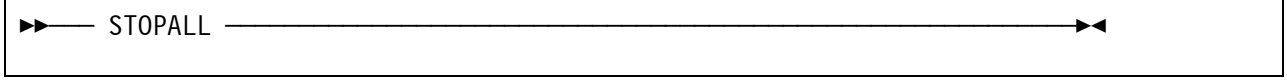
The STOPALL command stops all CPUs

### *7.106.2 Syntax*

**Descriptive**

```
STOPALL
```

**Diagram**




### *7.106.3 Parameter*

None.

### *7.106.4 Examples*

**Example 1:**

Stop all CPU's.

```
2007/06/30 14:26:19.137 00000FC4 stopall
```

**Figure 167: STOPALL command**

## 7.107 STORE (Store CPU status at absolute zero)

### *7.107.1 Function*

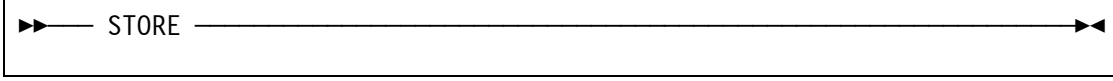
The STORE command stores the CPU status of the active CPU at absolute zero.

### *7.107.2 Syntax*

**Descriptive**

```
STORE
```

**Diagram**




### *7.107.3 Parameter*

None.

### *7.107.4 Examples*

**Example 1:**

Store status for active CPU.

```
2007/06/30 15:31:05.968 00000FC4 store
2007/06/30 15:31:05.968 00000FC4 HHCCP010I CPU0000 store status completed.
```

**Figure 168: STORE command**

## 7.108 SUSPEND (Suspend Hercules)

### *7.108.1 Function*

This command lets you suspend the current Hercules session and shutdown the host machine. Subsequently the suspended session can be resumed (see “RESUME” command). The data necessary to be saved for a later restart is saved in a packed (zipped) file called “hercules.srf.gz” which is located in the current configuration path.

After entering the SUSPEND command the CPUs are put in a stopped state and the contents of the main storage, CPU states, I/O device states and internal Hercules states are dumped onto the “Hercules.srf.gz” file. After the SUSPEND command has finished writing to the suspend file, it schedules an immediate shutdown of Hercules.

Currently device state is only fully saved for CKD disks. Each device class (e.g. TAPE, RDR, PUN, CTC) will need code to save and restore their state. Some states may not be possible to restore (e.g. active TCP/IP connections at the time of suspend). Currently the vector facility state is not saved, neither is the ECPSVM state.

The created suspend file is designed to be HERCULES release independent and to be host architecture independent. For example it is possible to take a suspend file created on HERCULES 3.05.0 on an Intel machine and resume on a Sun machine running HERCULES 3.07.0.

Please note that there are some caveats with suspend / resume processing which can affect the guest operating system running under Hercules. These caveats are described in the chapter about RESUME processing (**Error! Reference source not found.**, RESUME).

There are some caveats when suspending and resuming guest operating system processing:

- As seen by the guest operating system, the TOD clock will appear to jump a large value. Some guests may not cope very well with this. For example some guests may be dismayed because certain interrupts will occur way past its due time. Also for S/370 an interval timer interrupt may be lost if the guest is interrupted for more than half the Interval Timer wrap time (around 8 hours).
- Although some effort has been put in order to make this as transparent as possible (that is, it should appear to the guest operating system that the STOP key was pressed for a large amount of time), some state information may be missed.

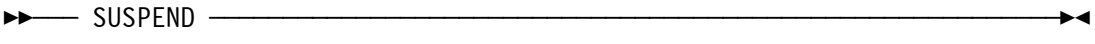
Some guest operating systems will fare better if the suspend state is prepared first. For MVS, as an example, it seems to help when a QUIESCE command and a SYSTEM RESTART manual operation are issued prior to suspend the system.

### *7.108.2 Syntax*

**Descriptive**

```
SUSPEND
```

**Diagram**



## 7.108.3 Parameter

None.

## 7.108.4 Examples

**Example 1:**

Suspend the Hercules session.

```
2007/06/30 14:15:41.328 00000FC4 suspend
2007/06/30 14:15:49.265 00000FC4 HHCIN900I Begin Hercules shutdown
2007/06/30 14:15:49.265 00000FC4 HHCIN901I Releasing configuration
2007/06/30 14:15:49.265 00000FC4 HHCCP008I CPU0001 thread ended: tid=00000F44, pid=1112
2007/06/30 14:15:49.265 00000458 HHCCP008I CPU0001 thread ended: tid=00000F44, pid=1112
```

**Figure 169: SUSPEND command**

## 7.109 SYNCIO (Display SYNCIO device statistics)

### *7.109.1 Function*

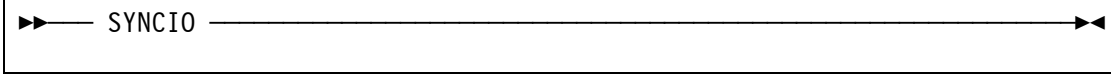
This command shows the SYNCIO device statistics for all DASD devices.

### *7.109.2 Syntax*

**Descriptive**

```
SYNCIO
```

**Diagram**




### *7.109.3 Parameter*

None.

### *7.109.4 Examples*

**Example 1:**

Display SYNCIO device statistics.

```
14:33:16.700 00000FC4 syncio
14:33:16.700 00000FC4 HHCPN072I 0130  synchronous:          397 asynchronous:           64
14:33:16.700 00000FC4 HHCPN072I 0131  synchronous:           11 asynchronous:            2
14:33:16.700 00000FC4 HHCPN072I 0132  synchronous:            7 asynchronous:            1
14:33:16.700 00000FC4 HHCPN072I 0133  synchronous:          134 asynchronous:           52
.
several lines not displayed
.
14:33:16.716 00000FC4 HHCPN072I 014B  synchronous:            7 asynchronous:            1
14:33:16.716 00000FC4 HHCPN072I 014C  synchronous:            7 asynchronous:            1
14:33:16.716 00000FC4 HHCPN072I 0160  synchronous:         2326 asynchronous:          629
14:33:16.716 00000FC4 HHCPN072I 0161  synchronous:           44 asynchronous:           19
14:33:16.716 00000FC4 HHCPN074I TOTAL synchronous:        57957 asynchronous:         9267   86%
```

**Figure 170: SYNCIO command**

## 7.110 SYSCLEAR (Issue SYSTEM CLEAR RESET manual operation)

### *7.110.1 Function*

The SYSCLEAR command issues a SYSTEM CLEAR RESET manual operation. This is basically the same as the SYSRESET command (see below), but also clears main storage to x'00' . Additionally the general purpose registers, control registers, etc. are reset to their initial value. At this point the system is essentially in the same state as it was just after having been started.

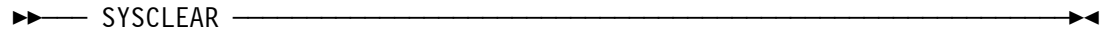
Please note that all CPUs must be stopped prior to issuing a system clear reset or the function will be rejected with message HHCPN053E.

### *7.110.2 Syntax*

**Descriptive**

```
SYSCLEAR
```

**Diagram**




### *7.110.3 Parameter*

None.

### *7.110.4 Examples*

**Example 1:**

CLEAR RESET the system.

```
2007/06/30 14:15:41.328 00000FC4 stopall
2007/06/30 14:15:51.221 00000FC4 sysclear
```

**Figure 171: SYSCLEAR command**

## 7.111 SYSRESET (Issue SYSTEM RESET manual operation)

### *7.111.1 Function*

The SYSRESET command issues a SYSTEM RESET manual operation. A CPU and I/O subsystem reset are performed.

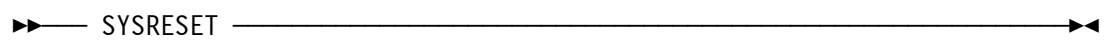
Please note that all CPUs must be stopped prior to issuing a system reset or the function will be rejected with message HHCPN053E.

### *7.111.2 Syntax*

**Descriptive**

SYSRESET

**Diagram**




### *7.111.3 Parameter*

None.

### *7.111.4 Examples*

**Example 1:**

RESET the system.

```
2007/06/30 14:16:42.228 00000FC4 stopall
2007/06/30 14:17:03.421 00000FC4 sysreset
```

**Figure 172: SYSRESET command**

## 7.112 T (Instruction trace)

### *7.112.1 Function*

This command sets the instruction tracing range. Instruction tracing is totally separate from the instruction stepping and breaking range. With or without a range the T command displays whether tracing is on or off and the range, if any.

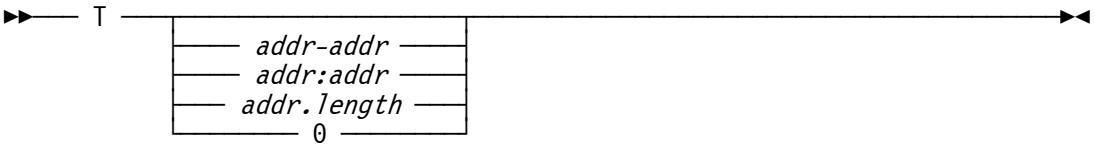
The T command by itself does not activate instruction tracing. Use the T+ command to activate instruction tracing. The command “T 0” eliminates the range (all addresses will be traced).

### *7.112.2 Syntax*

**Descriptive**

T [*addr-addr* | *addr:addr* | *addr.length* | 0]

**Diagram**




### *7.112.3 Parameter*

| | |
|---|---|
| **addr** | Specifies an address. The range can be coded as “addr–addr” or “addr:addr” (from begin address to end address) or as address.length (from begin address for length). |
| **length** | Specifies the length of the range. |
| **0** | No range is specified or an existing range is reset. Tracing is active for all addresses. |

### *7.112.4 Examples*

**Example 1:**

Set instruction tracing range.

```
2009/02/15 10:06:22.432 00000FC4 t 1048576-1056768
2009/02/15 10:06:22.432 00000FC4 HHCPN040I Instruction tracing off range 1048576-1056768
```

**Figure 173: T command**

## 7.113 T+ (Instruction trace on)

### *7.113.1 Function*

This command turns on instruction tracing. A range can be specified as for the T command. Instruction tracing is totally separate from the instruction stepping and breaking range.

After turning instruction tracing on, every instruction is listed with detailed trace information. The trace information includes the executing CPU, the PSW, the executed instruction in hexadecimal and in disassembled form as well as register hexadecimal displays of all involved register types.

This function is similar to the instruction stepping. The difference between these two function is that with instruction stepping the execution of every instruction has to be initiated by hitting ENTER on the Hercules console, whereas instruction tracing – once started – traces the instructions without interruption.

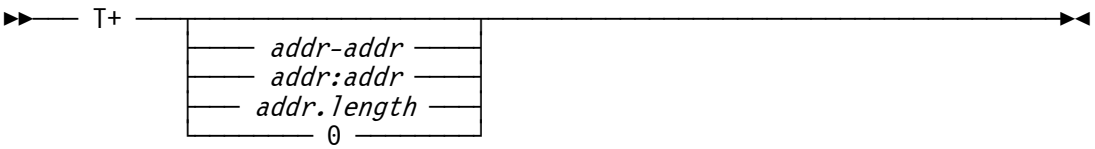
Please note that this function will create an enormous amount of output in the Hercules log. Running it on a relatively current machine (3 GHz HT Processor) for 10 seconds creates more than 10 MB of trace data.

### *7.113.2 Syntax*

**Descriptive**

T+ [*addr-addr* | *addr:addr* | *addr.length* | 0]

**Diagram**




### *7.113.3 Parameter*

**addr** Specifies an address. The range can be coded as “addr–addr” or “addr:addr” (from begin address to end address) or as address.length (from begin address for length).

**length** Specifies the length of the range.

**0** No range is specified or an existing range is reset. Tracing is active for all addresses.

## *7.113.4 Examples*

**Example 1:**

Turn instruction tracing on.

```
t+
HHCPN040I Instruction tracing on
CPU0000:  PSW=00080000 8000064C INST=50405000     ST    4,0(0,5)              store
CPU0000:  R:00002000:K:04=00000000 00000000 00000000 00000000  ................
CPU0000: GR00=00000000  GR01=00000000  GR02=00000000  GR03=00000032
CPU0000: GR04=00000000  GR05=00002000  GR06=00000000  GR07=00000000
CPU0000: GR08=00000000  GR09=00000000  GR10=00000000  GR11=00000000
CPU0000: GR12=00000000  GR13=00000000  GR14=00000000  GR15=00000000
CPU0000:  PSW=00080000 80000650 INST=1845         LR    4,5                   load_register
CPU0000: GR00=00000000  GR01=00000000  GR02=00000000  GR03=00000032
CPU0000: GR04=00000000  GR05=00002000  GR06=00000000  GR07=00000000
CPU0000: GR08=00000000  GR09=00000000  GR10=00000000  GR11=00000000
CPU0000: GR12=00000000  GR13=00000000  GR14=00000000  GR15=00000000
CPU0000:  PSW=00080000 80000652 INST=18F5         LR    15,5                  load_register
CPU0000: GR00=00000000  GR01=00000000  GR02=00000000  GR03=00000032
CPU0000: GR04=00002000  GR05=00002000  GR06=00000000  GR07=00000000
CPU0000: GR08=00000000  GR09=00000000  GR10=00000000  GR11=00000000
CPU0000: GR12=00000000  GR13=00000000  GR14=00000000  GR15=00000000
CPU0000:  PSW=00080000 8000065C INST=50400010     ST    4,16(0,0)             store
CPU0000:  R:00000010:K:04=00000000 00000390 00000000 00000000  ................
CPU0000: GR00=00000000  GR01=00000000  GR02=00000000  GR03=00000000
CPU0000: GR04=00033000  GR05=00033000  GR06=00000000  GR07=00000000
CPU0000: GR08=00000000  GR09=00000000  GR10=00000000  GR11=00000000
CPU0000: GR12=00000000  GR13=00000000  GR14=00000000  GR15=00033000
CPU0000:  PSW=00080000 80000660 INST=50400528     ST    4,1320(0,0)           store
CPU0000:  R:00000528:K:06=00000000 40000009 D840E5C3 D4D340D9  .... ...Q VCML R
CPU0000: GR00=00000000  GR01=00000000  GR02=00000000  GR03=00000000
CPU0000: GR04=00033000  GR05=00033000  GR06=00000000  GR07=00000000
CPU0000: GR08=00000000  GR09=00000000  GR10=00000000  GR11=00000000
CPU0000: GR12=00000000  GR13=00000000  GR14=00000000  GR15=00033000
CPU0000:  PSW=00080000 80000664 INST=58200010     L     2,16(0,0)             load
CPU0000:  R:00000010:K:06=00033000 00000390 00000000 00000000  ................
CPU0000: GR00=00000000  GR01=00000000  GR02=00000000  GR03=00000000
CPU0000: GR04=00033000  GR05=00033000  GR06=00000000  GR07=00000000
CPU0000: GR08=00000000  GR09=00000000  GR10=00000000  GR11=00000000
CPU0000: GR12=00000000  GR13=00000000  GR14=00000000  GR15=00033000
.
.
.
```

**Figure 174: T+ command**

## 7.114 T- (Instruction trace off)

### *7.114.1 Function*

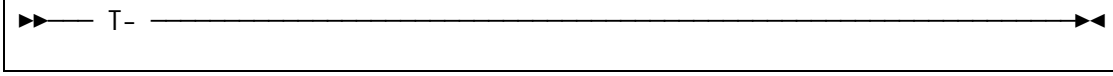
This command turns off the instruction tracing function.

### *7.114.2 Syntax*

**Descriptive**

T-

**Diagram**




### *7.114.3 Parameter*

None.

### *7.114.4 Examples*

**Example 1:**

Turn off instruction tracing.

```
2009/02/15 10:07:39.312 00000FC4 t-
2009/02/15 10:07:39.312 00000FC4 HHCPN040I Instruction tracing off range 1048576-1056768
```

**Figure 175: T- command**

## 7.115 T? (Instruction trace query)

### *7.115.1 Function*

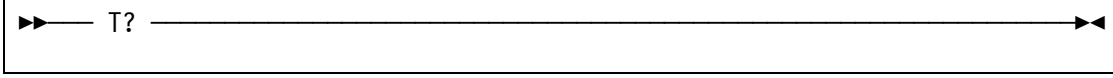
The T? command displays whether instruction tracing is on or off and the range if any.

### *7.115.2 Syntax*

**Descriptive**

T?

**Diagram**




### *7.115.3 Parameter*

None.

### *7.115.4 Examples*

**Example 1:**

Query instruction tracing.

```
2009/02/15 10:08:05.112 00000FC4 t?
2009/02/15 10:08:05.112 00000FC4 HHCPN040I Instruction tracing on range 1048576-1056768
```

**Figure 176: T? command**

## 7.116 T{+/-} CKD (Turn CKD_KEY tracing on or off)

### *7.116.1 Function*

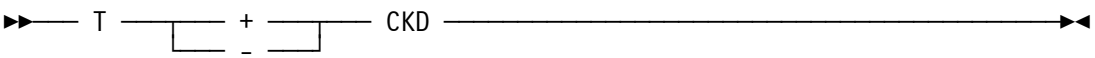
The T{+/-}CKD command turns the CKD key tracing on (T+CKD) or turns it off (T-CKD).

### *7.116.2 Syntax*

**Descriptive**

```
T{+ | -}CKD
```

**Diagram**




### *7.116.3 Parameter*

**T+CKD** The T+CKD turns on the CKD key trace. The plus sign and the CKD argument must immediately follow the T command (without an intervening blank).

**T-CKD** The T-CKD turns off the CKD key trace. The minus sign and theCKD argument must immediately follow the T command (without an intervening blank).

### *7.116.4 Examples*

**Example 1:**

Turning the CKD key trace on.

```
2007/06/01 05:52:41.290 00000C24 t+ckd
2007/06/01 05:53:02.634 00000C24 HHCDA055I search key SYS1.NUCLEUS
2007/06/01 05:53:02.650 00000C24 HHCDA055I search key IEANUC01
2007/06/01 05:53:02.775 00000C24 HHCDA055I search key IEAVNIPM
2007/06/01 05:53:02.775 00000C24 HHCDA055I search key IEAVNPA2
2007/06/01 05:53:02.822 00000C24 HHCDA055I search key IEAVNP01
2007/06/01 05:53:23.025 00000C24 HHCDA055I search key IEAVNPB2
2007/06/01 05:53:23.134 00000C24 HHCDA055I search key SYS1.LOGREC
2007/06/01 05:53:23.134 00000C24 HHCDA055I search key SYS1.SVCLIB
2007/06/01 05:53:23.134 00000C24 HHCDA055I search key IEAVNP11
2007/06/01 05:53:23.134 00000C24 HHCDA055I search key SYSCATLG
2007/06/01 05:53:23.134 00000C24 HHCDA055I search key Z9999994.VSAMDSPC
2007/06/01 05:53:23.150 00000C24 HHCDA055I search key IEAVNP1A
2007/06/01 05:53:23.165 00000C24 HHCDA055I search key IEAVNP03
```

```
2007/06/01 05:53:23.165 00000C24 HHCDA055I search key IEAVNP12
2007/06/01 05:53:23.165 00000C24 HHCDA055I search key SYS1.PARMLIB
2007/06/01 05:53:23.181 00000C24 HHCDA055I search key IEASYS00
2007/06/01 05:53:23.197 00000C24 HHCDA055I search key SYS1.LINKLIB
2007/06/01 05:53:23.197 00000C24 HHCDA055I search key LNKLST00
2007/06/01 05:53:23.197 00000C24 HHCDA055I search key SYS1.CMDLIB
2007/06/01 05:53:23.212 00000C24 HHCDA055I search key SYS2.LINKLIB
2007/06/01 05:53:23.228 00000C24 HHCDA055I search key SYS2.CMDLIB
2007/06/01 05:53:23.244 00000C24 HHCDA055I search key IEAVNP10
2007/06/01 05:53:23.259 00000C24 HHCDA055I search key IEAOPT00
2007/06/01 05:53:23.275 00000C24 HHCDA055I search key IEAIPS00
2007/06/01 05:53:23.306 00000C24 HHCDA055I search key ILRPREAD
2007/06/01 05:53:23.322 00000C24 HHCDA055I search key IEAVNP04
2007/06/01 05:53:23.322 00000C24 HHCDA055I search key ILROPS00
2007/06/01 05:53:23.415 00000C24 HHCDA055I search key IEAVNPA8
2007/06/01 05:53:23.415 00000C24 HHCDA055I search key IEAVNP05
2007/06/01 05:53:23.415 00000C24 HHCDA055I search key SYS1.LPALIB
2007/06/01 05:53:23.572 00000C24 HHCDA055I search key IEAPAK00
2007/06/01 05:53:28.587 00000C24 HHCDA055I search key IEABLD00
2007/06/01 05:53:28.603 00000C24 HHCDA055I search key ALLOC
2007/06/01 05:53:28.603 00000C24 HHCDA055I search key E
2007/06/01 05:53:28.603 00000C24 HHCDA055I search key HEWL
2007/06/01 05:53:28.603 00000C24 HHCDA055I search key IEWL
2007/06/01 05:53:28.603 00000C24 HHCDA055I search key IEWLF440
2007/06/01 05:53:28.603 00000C24 HHCDA055I search key IFOX01
2007/06/01 05:53:28.603 00000C24 HHCDA055I search key IFOX11
2007/06/01 05:53:28.603 00000C24 HHCDA055I search key IFOX62
2007/06/01 05:53:28.603 00000C24 HHCDA055I search key LINK
2007/06/01 05:53:28.619 00000C24 HHCDA055I search key ALLOC
2007/06/01 05:53:28.619 00000C24 HHCDA055I search key E
2007/06/01 05:53:28.619 00000C24 HHCDA055I search key LINK
2007/06/01 05:53:28.619 00000C24 HHCDA055I search key LOGOFF
2007/06/01 05:53:28.619 00000C24 HHCDA055I search key SUBMIT
2007/06/01 05:53:28.619 00000C24 HHCDA055I search key IEALOD00
```

**Figure 177: T+CKD command**

**Example 2:**

Turning the CKD key trace off.

```
2007/06/01 05:54:08.572 00000C24 t-ckd
2007/06/01 05:54:08.572 00000C24 HHCPN134I CKD KEY trace is now off
```

**Figure 178: T-CKD command**

## 7.117 T{+/-} dev (Turn CCW tracing on or off)

### *7.117.1 Function*

This command turns the CCW tracing for a certain device on (T+dev) or turns it off (T-dev). It is possible to trace several devices concurrently. To enable the tracing of more than one device the T+ command must be issued for each device. The traces can be turned off independently of each other.

This function is similar to the CCW stepping described in section 7.88. The difference between these two function is that with CCW stepping the execution of every CCW has to be initiated by hitting ENTER on the Hercules console, whereas CCW tracing – once started – traces the instructions without interruption.

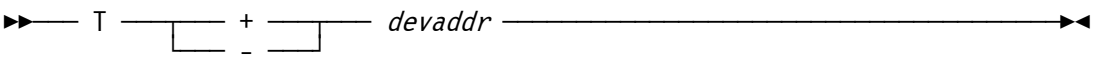
Please note that the tracing function will create an enormous amount of output in the Hercules log.

### *7.117.2 Syntax*

**Descriptive**

T{+ | -}*devaddr*

**Diagram**




### *7.117.3 Parameter*

**devaddr** This is the address of the device for which the CCW tracing is to be turned on or off.

**+** The plus sign turns on the CCW tracing for the given device. The plus sign must imme-- diately follow the T command (without an intervening blank).

**-** The minus sign turns off the CCW tracing for the given device. The minus sign must - immediately follow the T command (without an intervening blank).

### *7.117.4 Examples*

**Example 1:**

Turn on the CCW tracing for device 0148.

```
t+0148
HHCPN136I CCW tracing is now on for device 0148
0148:start i/o file[0] bufcur -1 cache[-1]
HHCCP048I 0148:CCW=02000000 60000018=>02000000 60000018 00000000 00000000 ....-.....
0148:HHCDA038I seeking to cyl 0 head 0
0148:read  trk   0 (asynchronous)
```

```
0148:0 rdtrk     0
0148:0 rdtrk[0] 0 cache miss
0148:0 rdtrk[0] 0 buf 00DA4FA0 len 56832
0148:trk[0] read_trkimg
0148:file[0] l2[0,0] trk[0] read_l2ent 0x710
0148:file[0] read_l2 0 active -1 -1 -1
0148:l2[0,0] cache[0] miss
0148:file[0] fd[5] read, off 0x710 len 2048
0148:file[0] cache[0] l2[0] read offset 0x710
0148:file[0] l2[0,0] trk[0] read_l2ent 0xf10 3057 3057
0148:file[0] fd[5] read, off 0xf10 len 3057
0148:0 rdtrk[0] 0 complete buf 00DA4FA0:0100000000
0148:uncompress comp 1 len 3057 maxlen 56832 trk 0
0148:newbuf malloc 00DBCEC8 len 56832
0148:uncompress zlib newlen 6491 rc 0
0148:validating trk 0 len 6491 0000000000 0000000000000008
0148:read  trk   0 uncompressed len 6491
0148:HHCDA041I read count orientation is index
0148:HHCDA043I cyl 0 head 0 record 0 kl 0 dl 8 of 0
0148:HHCDA043I cyl 0 head 0 record 1 kl 4 dl 24 of 0
0148:HHCDA045I read data 24 bytes
HHCCP075I 0148:Stat=0C00 Count=0000 =>00080000 8000015A 06004A98 60000060 .......!...q-..-
HHCCP048I 0148:CCW=06004A98 60000060=>00000000 00000000 00000000 00000000 ................
0148:HHCDA041I read count orientation is data
0148:HHCDA043I cyl 0 head 0 record 2 kl 4 dl 144 of 0
0148:HHCDA045I read data 144 bytes
.
.
.
0148:HHCDA041I read count orientation is count
0148:HHCDA043I cyl 0 head 0 record 1 kl 4 dl 24 of 0
HHCCP075I 0148:Stat=0C00 Count=0000
HHCCP048I 0148:CCW=08004AA0 00000000=>31004ABE 40000005 08004AA0 00000000 .... ..........
HHCCP048I 0148:CCW=31004ABE 40000005=>00000000 04000000 00000000 00000000 ................
0148:HHCDA041I read count orientation is count
0148:HHCDA043I cyl 0 head 0 record 2 kl 4 dl 144 of 0
HHCCP075I 0148:Stat=0C00 Count=0000
HHCCP048I 0148:CCW=08004AA0 00000000=>31004ABE 40000005 08004AA0 00000000 .... ..........
HHCCP048I 0148:CCW=31004ABE 40000005=>00000000 04000000 00000000 00000000 ................
0148:HHCDA041I read count orientation is count
0148:HHCDA043I cyl 0 head 0 record 3 kl 4 dl 80 of 0
HHCCP075I 0148:Stat=0C00 Count=0000
HHCCP048I 0148:CCW=08004AA0 00000000=>31004ABE 40000005 08004AA0 00000000 .... ..........
HHCCP048I 0148:CCW=31004ABE 40000005=>00000000 04000000 00000000 00000000 ................
0148:HHCDA041I read count orientation is count
0148:HHCDA043I cyl 0 head 0 record 4 kl 0 dl 6170 of 0
HHCCP075I 0148:Stat=4C00 Count=0000
HHCCP048I 0148:CCW=06000000 20002F55=>00080000 8000015A 06004A98 60000060 .......!...q-..-
0148:HHCDA045I read data 6170 bytes
HHCCP075I 0148:Stat=0C00 Count=173B =>00080000 800005EC 40404040 F0F04040 ........    00
0148:end i/o bufcur 0 cache[0] waiters 0
```

**Figure 179: T+ dev command**

**Example 2:**

Turn off the CCW tracing for device 0148.

```
2007/06/30 14:21:05.156 00000FC4 t-0148
2007/06/30 14:21:05.156 00000FC4 HHCPN136I CCW tracing is now off for device 0148
```

**Figure 180: T- dev command**

**Example 3:**

Turn on the CCW tracing for several devices.

```
2007/06/30 14:23:15.246 00000FC4 t+0148
2007/06/30 14:23:15.246 00000FC4 HHCPN136I CCW tracing is now on for device 0148
2007/06/30 14:23:15.246 00000FC4 t+0561
2007/06/30 14:23:15.246 00000FC4 HHCPN136I CCW tracing is now on for device 0561
```

**Figure 181: T+ dev command (several devices)**

# 7.118 TIMERINT (Display or set timers update interval)

## 7.118.1 Function

The TIMERINT parameter displays or sets the internal timer update interval in microseconds. This parameter specifies how frequently Hercules's internal timers-update thread updates the TOD clock, CPU Timer and other architectural related clock and timer values.

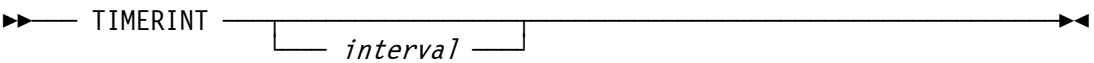
The default interval is 50 microseconds which attempts to strike a reasonable balance between clock accuracy and overall host performance.

## 7.118.2 Syntax

**Descriptive**

```
TIMERINT [interval]
```

**Diagram**




## 7.118.3 Parameter

**interval** Specifies the timer update interval in microseconds. The minimum allowed value for the interval is 1 microsecond and the maximum is 1'000'000 microseconds (one second).

CAUTION: While a lower TIMERINT value may help increase the accuracy of the guest's TOD clock and CPU Timer values, it could also have severe negative impact on the overall performance of the host operating system. This is especially true when a low TIMERINT value is coupled with a high HERCPRIO and TODPRIO priority setting.

Exercise extreme caution when choosing your desired TIMERINT in relationship to your chosen HERCPRIO and TODPRIO priority settings.

## 7.118.4 Examples

**Example 1:**

Display current timers update interval.

```
2007/06/30 14:04:44.578 00000FC4 timerint
2007/06/30 14:04:44.578 00000FC4 HHCPN037I Timer update interval = 50 microsecond(s)
```

**Figure 182: TIMERINT command (list current value)**

**Example 2:**

Set a new timers update interval.

```
2007/06/30 14:04:55.296 00000FC4 timerint 100
2007/06/30 14:04:55.296 00000FC4 HHCPN037I Timer update interval = 100 microsecond(s)
```

**Figure 183: TIMERINT command (set new value)**

## 7.119 TLB (Display TLB tables)

### *7.119.1 Function*

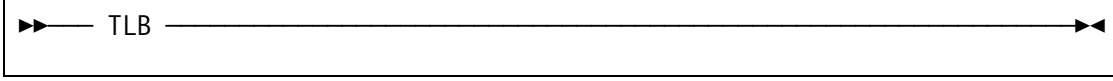
The TLB command shows the actual contents of the TLB (Translation Lookaside Buffer) tables.

### *7.119.2 Syntax*

**Descriptive**

TLB

**Diagram**




### *7.119.3 Parameter*

None.

### *7.119.4 Examples*

**Example 1:**

Display the TLB tables.

```
00000FC4 tlb
00000FC4 tlbID 0x0011c4 mainstor 10041000
00000FC4   ix              asd            vaddr              pte   id c p r w ky       main
00000FC4 *000 000000003fffe07f 0000000000000000 0000000000000000 11c4 0 0 0 0 00 3fbe3000
00000FC4 *001 00000000022b907f 0000000001801000 000000003e91e008 11c4 1 0 0 0 00 3f11d000
00000FC4 *002 00000000022b907f 0000000001802000 000000003e91f008 11c4 1 0 0 0 00 3f121000
.
several lines not displayed
.
00000FC4 *3fd 000000000a9eb07f 00000000013fd000 000000003f760208 11c4 1 1 0 0 00 3e41b000
00000FC4 *3fe 000000000a9eb07f 000000007fffe000 000000000aef4000 11c4 0 0 0 0 00 5508a000
00000FC4 *3ff 000000000a9eb07f 0000000000fff000 000000003f362208 11c4 1 1 0 0 00 3fc1b000
00000FC4 792 tlbID matches
```

**Figure 184: TLB command**

## 7.120 TODDRAG (Display or set TOD clock drag factor)

### 7.120.1 Function

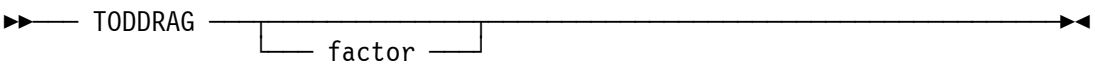
The TODDRAG command displays the current setting of the TOD clock drag factor or sets a new value. This parameter can be used to slow down or speed up the TOD clock by a factor of *nn.* A significant slow-down can improve the performance of some operating systems which consume large amounts of CPU time processing timer interrupts. A drag factor of 2.0 slows down the clock by 50%, a drag factor of 0.5 doubles the speed of the clock, a drag factor of 1.01 slows down the clock by 1% and 0.99 speeds up the clock by 1%.

### 7.120.2 Syntax

**Descriptive**

```
TODDRAG [factor]
```

**Diagram**




### 7.120.3 Parameter

**factor** The factor to which the TOD clock has to be slowed down or sped up.

### 7.120.4 Examples

**Example 1:**

Display current TOD clock drag factor.

```
2007/06/30 14:33:29.653 00000FC4 toddrag
2007/06/30 14:33:29.653 00000FC4 HHCPN036I TOD clock drag factor = 1.000000
```

**Figure 185: TODDRAG command (display TOD clock factor)**

**Example 2:**

Set the TOD clock drag factor.

```
2007/06/30 14:33:29.653 00000FC4 toddrag 0.99
2007/06/30 14:33:29.653 00000FC4 HHCPN036I TOD clock drag factor = 0.990000
```

**Figure 186: TODDRAG command (set TOD clock factor)**

## 7.121 TRACEOPT (Instruction trace display options)

### *7.121.1 Function*

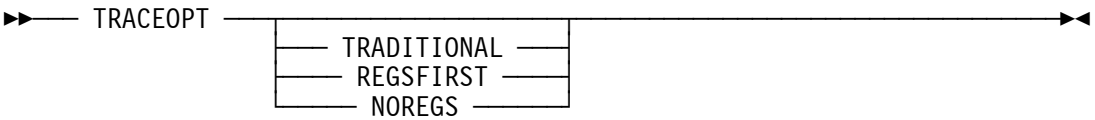
The TRACEOPT command determines how the registers are displayd during instruction tracing and stepping. Entering the command without any argument simply displays the current trace mode.

### *7.121.2 Syntax*

**Descriptive**

```
TRACEOPT [TRADITIONAL | REGSFIRST | NOREGS]
```

**Diagram**




### *7.121.3 Parameter*

**TRADITIONAL** Displays the registers following the instruction about to be executed such that pressing enter (to execute the displayed instruction) then shows the next instruction to be executed followed by the updated registers display.

**REGSFIRST** Displays the current register contents followed by the instruction about to be executed such that pressing enter (to execute the displayed instruction) then shows the updated registers followed by the next instruction to be executed.

**NOREGS** Suppresses the registers display altogether and shows just the instruction to be executed.

## *7.121.4 Examples*

**Example 1:**

Display the instruction trace in traditional mode (instruction followed by registers).

```
traceopt traditional
HHCPN162I Hercules instruction trace displayed in traditional mode

CPU0000:  PSW=00080000 800005EC INST=B7000444     LCTL  0,0,1092(0)            load_control
CPU0000:  R:00000444:K:06=01B00200 0000007F FE000000 00000001  .......".........
CPU0000: GR00=00000000  GR01=00000000  GR02=00000000  GR03=00000000
CPU0000: GR04=00000000  GR05=00000000  GR06=00000000  GR07=00000000
CPU0000: GR08=00000000  GR09=00000000  GR10=00000000  GR11=00000000
CPU0000: GR12=00000000  GR13=00000000  GR14=00000000  GR15=00000000
```

**Figure 187: TRACEOPT command (TRADITIONAL)**

**Example 2:**

Display the instruction trace in regsfirst mode (registers followed by instruction).

```
traceopt regsfirst
HHCPN162I Hercules instruction trace displayed in regsfirst mode

CPU0000: GR00=00000000  GR01=00000000  GR02=00000000  GR03=00000000
CPU0000: GR04=00000000  GR05=00000000  GR06=00000000  GR07=00000000
CPU0000: GR08=00000000  GR09=00000000  GR10=00000000  GR11=00000000
CPU0000: GR12=00000000  GR13=00000000  GR14=00000000  GR15=00000000
CPU0000:  PSW=00080000 800005F4 INST=A7F40002     BRC   15,4        branch_relative_on_condition
```

**Figure 188: TRACEOPT command (REGSFIRST)**

**Example 3:**

Display the instruction trace in noregs mode (instruction only).

```
traceopt noregs
HHCPN162I Hercules instruction trace displayed in noregs mode

CPU0000:  PSW=00080000 800005FC INST=D207006804B8 MVC   104(8,0),1208(0)       move_character
CPU0000:  R:00000068:K:06=000A0000 0002007B 00000000 00000000  .......#........
CPU0000:  R:000004B8:K:06=00080000 80000606 20000000 00100000  ................
CPU0000:  PSW=00080000 80000602 INST=B2790000     SACF  0(0)      set_address_space_control_fast
CPU0000:  R:00000000:K:06=00080000 800005EC 40404040 F0F04040  ........    00
```

**Figure 189: TRACEOPT command (NOREGS)**

## 7.122 TT32 (Control / query CTCI-WIN functionality)

### *7.122.1 Function*

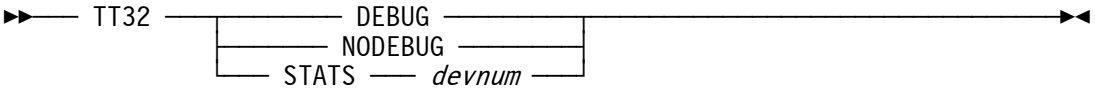
The TT32 command controls or queries the CTCI-WIN functionality for a given device address. It allows you to enable or disable global CTCI-WIN debug tracing, or displays TunTap32 statistics for the specified CTC device.

### *7.122.2 Syntax*

**Descriptive**

```
TT32 {DEBUG | NODEBUG | STATS devnum}
```

**Diagram**




### *7.122.3 Parameter*

**DEBUG** Enable global CTCI-WIN debug tracing.

**NODEBUG** Disable global CTCI-WIN debug tracing.

**STATS** Display TunTap32 statistics for CTC device.

**devnum** Device number of the device for which the TunTap32 statistics have to displayed.

### *7.122.4 Examples*

**Example 1:**

Enable global CTCI-WIN debug tracing.

```
2007/06/30 17:29:16.296 00000FC4 tt32 debug
2007/06/30 17:29:16.296 00000FC4 HHCPNxxxI TT32 debug tracing messages enabled
```

**Figure 190: TT32 DEBUG command**

**Example 2:**

Disable global CTCI-WIN debug tracing.

```
2007/06/30 17:29:16.296 00000FC4 tt32 nodebug
2007/06/30 17:29:16.296 00000FC4 HHCPNxxxI TT32 debug tracing messages disabled
```

**Figure 191: TT32 NODEBUG command**

**Example 3:**

Display TunTap32 statistics for device 0E20.

```
2007/06/30 14:33:48.762 00000FC4 tt32 stats 0e20
2007/06/30 14:33:48.762 00000FC4 TunTap32.dll Statistics:
2007/06/30 14:33:48.762 00000FC4
2007/06/30 14:33:48.762 00000FC4 Size of Kernel Hold Buffer:       1024K
2007/06/30 14:33:48.762 00000FC4 Size of DLL I/O Buffer:             64K
2007/06/30 14:33:48.762 00000FC4 Maximum DLL I/O Bytes Received:      4K
2007/06/30 14:33:48.762 00000FC4
2007/06/30 14:33:48.762 00000FC4            0 Write Calls
2007/06/30 14:33:48.762 00000FC4            0 Write I/Os
2007/06/30 14:33:48.762 00000FC4          346 Read Calls
2007/06/30 14:33:48.762 00000FC4         1770 Read I/Os
2007/06/30 14:33:48.762 00000FC4          950 Packets Read
2007/06/30 14:33:48.762 00000FC4            0 Packets Written
2007/06/30 14:33:48.762 00000FC4       524372 Bytes Read
2007/06/30 14:33:48.762 00000FC4            0 Bytes Written
2007/06/30 14:33:48.762 00000FC4            0 Internal Packets
2007/06/30 14:33:48.762 00000FC4          913 Ignored Packets
```

**Figure 192: TT32 STATS command**

## 7.123 U (Disassemble storage)

### 7.123.1 Function

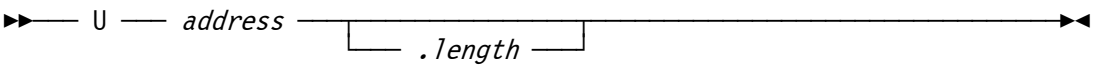
The U command lets you disassemble storage. The argument specifies the storage address to be disassembled. The length of the storage area can be given as an optional argument. The listed storage area is formatted as one line per instruction. Every instruction is shown as a hexadecimal string and as readable assembler statement. Storage areas that can not be disassembled (data areas) are marked with question marks.

### 7.123.2 Syntax

**Descriptive**

U *address* [*.length*]

**Diagram**




### 7.123.3 Parameter

**address** The address of the storage area that is to be disassembled.

**length** The length of the storage that is to be disassembled. This value must be specified in hexadecimal. If the length is omitted then 64 bytes of storage are disassembled by default. The length argument must follow immediately the address parameter, seperated by a dot (“address.length”).

### 7.123.4 Examples

**Example 1:**

Disassemble 4096 bytes beginning at address 00040130.

```
00000C24 u 00040130.1000
00000C24 P00040130: 90DF02EC     STM   13,15,748(0)          store_multiple
00000C24 P00040134: 4166000C     LA    6,12(6,0)             load_address
00000C24 P00040138: 58606160     L     6,352(0,6)            load
00000C24 P0004013C: 58300010     L     3,16(0,0)             load
00000C24 P00040140: 58430184     L     4,388(3,0)            load
00000C24 P00040144: 55C06058     CL    12,88(0,6)            compare_logical
00000C24 P00040148: 478060E0     BC    8,224(0,6)            branch_on_condition
00000C24 P0004014C: 8840000A     SRL   4,10(0)               shift_right_single_logical
```

```
00000C24 P00040150: 47F060E4     BC    15,228(0,6)           branch_on_condition
00000C24 P00040154: 88400005     SRL   4,5(0)                shift_right_single_logical
00000C24 P00040158: 9620820C     OI    524(8),32             or_immediate
00000C24 P0004015C: 55C06058     CL    12,88(0,6)            compare_logical
00000C24 P00040160: 478060F8     BC    8,248(0,6)            branch_on_condition
00000C24 P00040164: 58506228     L     5,552(0,6)            load
00000C24 P00040168: 47F060FC     BC    15,252(0,6)           branch_on_condition
00000C24 P0004016C: 5850622C     L     5,556(0,6)            load
00000C24 P00040170: 58AB0000     L     10,0(11,0)            load
00000C24 P00040174: 12AA         LTR   10,10                 load_and_test_register
00000C24 P00040176: 4770611A     BC    7,282(0,6)            branch_on_condition
00000C24 P0004017A: 94DF820C     NI    524(8),223            and_immediate
00000C24 P0004017E: 9102022A     TM    554(0),2              test_under_mask
00000C24 P00040182: 0717         BCR   1,7                   branch_on_condition_register
00000C24 P00040184: 980402B8     LM    0,4,696(0)            load_multiple
00000C24 P00040188: 98DF02EC     LM    13,15,748(0)          load_multiple
00000C24 P0004018C: 07F7         BCR   15,7                  branch_on_condition_register
00000C24 P0004018E: 465060FC     BCT   5,252(0,6)            branch_on_count
00000C24 P00040192: 55A06230     CL    10,560(0,6)           compare_logical
00000C24 P00040196: 4740620E     BC    4,526(0,6)            branch_on_condition
00000C24 P0004019A: 48E301DC     LH    14,476(3,0)           load_halfword
00000C24 P0004019E: 54A06234     N     10,564(0,6)           and
00000C24 P000401A2: 15AE         CLR   10,14                 compare_logical_register
00000C24 P000401A4: 4720620E     BC    2,526(0,6)            branch_on_condition
00000C24 P000401A8: B60082B0     STCTL 0,0,688(8)            store_control
00000C24 P000401AC: B6000308     STCTL 0,0,776(0)            store_control
00000C24 P000401B0: D40303086166 NC    776(4,0),358(6)       and_character
00000C24 P000401B6: D6030308616A OC    776(4,0),362(6)       or_character
00000C24 P000401BC: B7000308     LCTL  0,0,776(0)            load_control
00000C24 P000401C0: AD010308     STOSM 776(0),1              store_then_or_system_mask
00000C24 P000401C4: ACFE0308     STNSM 776(0),254            store_then_and_system_mask
00000C24 P000401C8: B6000308     STCTL 0,0,776(0)            store_control
00000C24 P000401CC: D60082B00308 OC    688(1,8),776(0)       or_character
00000C24 P000401D2: B70082B0     LCTL  0,0,688(8)            load_control
00000C24 P000401D6: 47F0616E     BC    15,366(0,6)           branch_on_condition
.
.
.
remaining lines not displayed
.
.
.
```

**Figure 193: U command**

## 7.124 UPTIME (Display Hercules Emulator uptime)

### *7.124.1 Function*

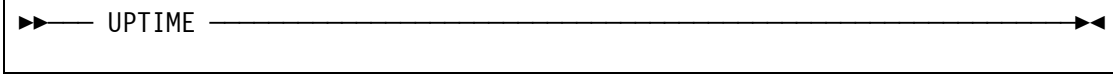
The UPTIME console command displays how long the Hercules Emulator has been running.

### *7.124.2 Syntax*

**Descriptive**

```
UPTIME
```

**Diagram**




### *7.124.3 Parameter*

**None.**

### *7.124.4 Examples*

**Example 1:**

Display how long the Hercules Emulator has been running so far (uptime < 1 day).

```
2009/06/15 01:02:53.500 00000C54 uptime
2009/06/15 01:02:53.500 00000C54 Hercules has been up for 06:33:12.
```

**Figure 194: UPTIME command (uptime < 1 day).**

**Example 2:**

Display how long the Hercules Emulator has been running so far (uptime > 1 week).

```
2009/06/15 01:02:53.500 00000C54 uptime
2009/06/15 01:02:53.500 00000C54 Hercules has been up for 3 weeks, 5 days, 16:48:15.
```

**Figure 195: UPTIME command (uptime > 1 week).**

## 7.125 V (Display or alter virtual storage)

### *7.125.1 Function*

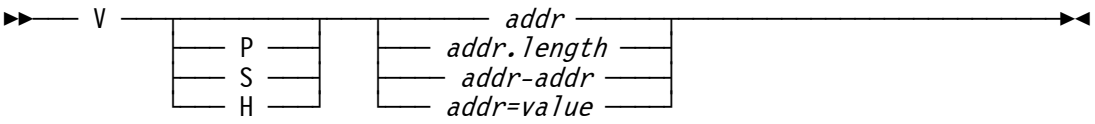
The V command displays or alters virtual storage.

### *7.125.2 Syntax*

**Descriptive**

V [P | S | H] {*addr* | *addr.length* | *addr-addr* | *addr=value*}

**Diagram**




### *7.125.3 Parameter*

**addr** Address of the virtual storage area that is to be displayed or altered.

**length** Length of the storage area that is to be displayed. The value of *.len* must be given in hexadecimal. If the length is omitted then 64 bytes of virtual storage are displayed by default.

**P** The optional 'P' argument will force Primary translation instead of current PSW mode.

**S** The optional 'S' argument will force Secondary translation instead of current PSW mode.

**H** The optional 'H' argument will force Home translation instead of current PSW mode.

**addr-addr** Address range (from – to) of the virtual storage that is to be displayed.

**value** Value is a hex-string of up to 32 pairs of digits (32 bytes) which are to be written to the virtual storage address given by the *addr* parameter. After altering the storage 16 bytes of virtual storage starting at *addr* are displayed.

## *7.125.4 Examples*

**Example 1:**

Display 256 (x'FF') bytes of virtual storage starting from address x'00010000'.

```
05:56:20.087 00000C24 v 00010000.ff
05:56:20.087 00000C24 V:00010000 (primary) R:00010000
05:56:20.087 00000C24 V:00010000:K:04=4770A1C8 5820901C 47F0A1F0 91802003  ..~H.....0~0j...
05:56:20.087 00000C24 V:00010010:K:04=47E0A1EC 58202004 47F0A1F0 96803200  .\~......0~0o...
05:56:20.087 00000C24 V:00010020:K:04=91803200 4770A1FE 12224770 A1DC5850  j.....~.....~..&
05:56:20.087 00000C24 V:00010030:K:04=30201FCC 195C4770 A01C50C0 30245820  .....*....&{....
05:56:20.087 00000C24 V:00010040:K:04=30089140 320047E0 A28441B0 202C58D0  ..j ...\sd.....}
05:56:20.087 00000C24 V:00010050:K:04=02FC98CD D19005ED 58C031F4 12CC4780  ..q.J....{.4....
05:56:20.087 00000C24 V:00010060:K:04=A24E9120 300047E0 A2424160 203047F0  s+j....\s..-...0
05:56:20.087 00000C24 V:00010070:K:04=A2464160 20485870 31F445E0 A2E458C0  s..-.....4.\sU.{
05:56:20.087 00000C24 V:00010080:K:04=31F812CC 4780A262 187C4160 203045E0  .8....s..@.-...\
05:56:20.087 00000C24 V:00010090:K:04=A2E458C0 31FC12CC 4780A276 187C4160  sU.{......s..@.-
05:56:20.087 00000C24 V:000100A0:K:04=204045E0 A2E441B0 202C58D0 02FC98CD  . .\sU.....}..q.
05:56:20.087 00000C24 V:000100B0:K:04=D19C05ED 91803001 4770A2DA 1FCC59C0  J...j.....s....{
05:56:20.087 00000C24 V:000100C0:K:04=30504780 A2A691C0 32004770 A2A659C0  .&..swj{....sw.{
05:56:20.087 00000C24 V:000100D0:K:04=200C4780 A2DA58C0 305412CC 4770A2DA  ....s..{......s.
05:56:20.087 00000C24 V:000100E0:K:04=1F114180 0001BA18 30544770 A2DA5810  ............s...
05:56:20.087 00000C24 V:000100F0:K:04=305858F0 001058F0 F2645800 F0005000  ...0...02...0.&.
```

**Figure 196: V command (display virtual storage with length)**

**Example 2:**

Change virtual storage at address x'00010000' to x'FFFFFFFF'.

```
05:58:33.728 00000C24 v 00010000=FFFFFFFF
05:58:33.728 00000C24 V:00010000 (primary) R:00010000
05:58:33.728 00000C24 V:00010000:K:06=FFFFFFFF 5820901C 47F0A1F0 91802003  .........0~0j...
```

**Figure 197: V command (alter virtual storage)**

**Example 3:**

Display virtual storage from address x'00010000' to adress x'00010100'.

```
05:59:14.447 00000C24 v 00010000-00010100
05:59:14.447 00000C24 V:00010000 (primary) R:00010000
05:59:14.447 00000C24 V:00010000:K:06=FFFFFFFF 5820901C 47F0A1F0 91802003  .........0~0j...
05:59:14.447 00000C24 V:00010010:K:06=47E0A1EC 58202004 47F0A1F0 96803200  .\~......0~0o...
05:59:14.447 00000C24 V:00010020:K:06=91803200 4770A1FE 12224770 A1DC5850  j.....~.....~..&
05:59:14.447 00000C24 V:00010030:K:06=30201FCC 195C4770 A01C50C0 30245820  .....*....&{....
05:59:14.447 00000C24 V:00010040:K:06=30089140 320047E0 A28441B0 202C58D0  ..j ...\sd.....}
05:59:14.447 00000C24 V:00010050:K:06=02FC98CD D19005ED 58C031F4 12CC4780  ..q.J....{.4....
05:59:14.447 00000C24 V:00010060:K:06=A24E9120 300047E0 A2424160 203047F0  s+j....\s..-...0
05:59:14.447 00000C24 V:00010070:K:06=A2464160 20485870 31F445E0 A2E458C0  s..-.....4.\sU.{
```

```
05:59:14.447 00000C24 V:00010080:K:06=31F812CC 4780A262 187C4160 203045E0  .8....s..@.-...\
05:59:14.447 00000C24 V:00010090:K:06=A2E458C0 31FC12CC 4780A276 187C4160  sU.{......s..@.-
05:59:14.447 00000C24 V:000100A0:K:06=204045E0 A2E441B0 202C58D0 02FC98CD  . .\sU.....}..q.
05:59:14.447 00000C24 V:000100B0:K:06=D19C05ED 91803001 4770A2DA 1FCC59C0  J...j.....s....{
05:59:14.447 00000C24 V:000100C0:K:06=30504780 A2A691C0 32004770 A2A659C0  .&..swj{....sw.{
05:59:14.447 00000C24 V:000100D0:K:06=200C4780 A2DA58C0 305412CC 4770A2DA  ....s..{......s.
05:59:14.447 00000C24 V:000100E0:K:06=1F114180 0001BA18 30544770 A2DA5810  ............s...
05:59:14.447 00000C24 V:000100F0:K:06=305858F0 001058F0 F2645800 F0005000  ...0...02...0.&.
05:59:14.447 00000C24 V:00010100:K:06=1004BA01 F0004770 A2CE982E 31C0947F  ....0...s.q..{m"
```

**Figure 198: V command (display virtual storage with range)**

**Example 4:**

Display 256 (x'FF') bytes of virtual storage starting from address x'00010000'. Force Secondary translation instead of current PSW mode.

```
05:56:20.087 00000C24 v s 00010000.ff
05:56:20.087 00000C24 V:00010000 (secondary) R:00010000
05:56:20.087 00000C24 V:00010000:K:04=4770A1C8 5820901C 47F0A1F0 91802003  ..~H.....0~0j...
05:56:20.087 00000C24 V:00010010:K:04=47E0A1EC 58202004 47F0A1F0 96803200  .\~......0~0o...
05:56:20.087 00000C24 V:00010020:K:04=91803200 4770A1FE 12224770 A1DC5850  j.....~.....~..&
05:56:20.087 00000C24 V:00010030:K:04=30201FCC 195C4770 A01C50C0 30245820  .....*....&{....
05:56:20.087 00000C24 V:00010040:K:04=30089140 320047E0 A28441B0 202C58D0  ..j ...\sd.....}
05:56:20.087 00000C24 V:00010050:K:04=02FC98CD D19005ED 58C031F4 12CC4780  ..q.J....{.4....
05:56:20.087 00000C24 V:00010060:K:04=A24E9120 300047E0 A2424160 203047F0  s+j....\s..-...0
05:56:20.087 00000C24 V:00010070:K:04=A2464160 20485870 31F445E0 A2E458C0  s..-.....4.\sU.{
05:56:20.087 00000C24 V:00010080:K:04=31F812CC 4780A262 187C4160 203045E0  .8....s..@.-...\
05:56:20.087 00000C24 V:00010090:K:04=A2E458C0 31FC12CC 4780A276 187C4160  sU.{......s..@.-
05:56:20.087 00000C24 V:000100A0:K:04=204045E0 A2E441B0 202C58D0 02FC98CD  . .\sU.....}..q.
05:56:20.087 00000C24 V:000100B0:K:04=D19C05ED 91803001 4770A2DA 1FCC59C0  J...j.....s....{
05:56:20.087 00000C24 V:000100C0:K:04=30504780 A2A691C0 32004770 A2A659C0  .&..swj{....sw.{
05:56:20.087 00000C24 V:000100D0:K:04=200C4780 A2DA58C0 305412CC 4770A2DA  ....s..{......s.
05:56:20.087 00000C24 V:000100E0:K:04=1F114180 0001BA18 30544770 A2DA5810  ............s...
05:56:20.087 00000C24 V:000100F0:K:04=305858F0 001058F0 F2645800 F0005000  ...0...02...0.&.
```

**Figure 199: V command (display virtual storage with 'S' option)**

## 7.126 VERSION (Display version information)

### *7.126.1 Function*

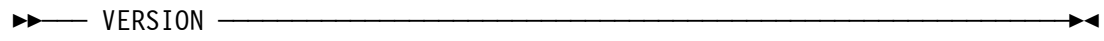
The version command is used to display various version and build information about the Hercules emulator.

### *7.126.2 Syntax*

**Descriptive**

```
VERSION
```

**Diagram**




### *7.126.3 Parameter*

None.

### *7.126.4 Examples*

**Example 1:**

Display the Hercules version information.

```
06:00:00.025 00000C24 version
06:00:00.025 00000C24 Hercules Version 3.06
06:00:00.025 00000C24 (c)Copyright 1999-2007 by Roger Bowler, Jan Jaeger, and others
06:00:00.025 00000C24 Built on Jan 10 2009 at 01:56:22
06:00:00.025 00000C24 Build information:
06:00:00.025 00000C24   Win32 (MSVC) build
06:00:00.025 00000C24   Modes: S/370 ESA/390 z/Arch
06:00:00.025 00000C24   Max CPU Engines: 8
.
.
.
06:00:00.025 00000C24   Automatic Operator support
06:00:00.025 00000C24   Machine dependent assists: cmpxchg1 cmpxchg4 cmpxchg8 fetch_dw store_dw
06:00:00.025 00000C24 Running on Micky Windows_NT-5.1 i686 MP=2
```

**Figure 200: VERSION command**

# 8. Hercules Utilities

## 8.1 Overview

Several utility programs exist to support the Hercules Emulator. Most of them are used when working with DASD and TAPE files. Other utilites perform some special functions with configuration files.

All these utility programs are pure line-command mode programs (i.e. Windows DOS or Unix/Linux shell). They can be called manually from the command line prompt or built into batch files. This is generally the preferred method due to its greater flexibility and automation possibilities.

If the Hercules Windows GUI (HercWinGUI) is used the utilities can also be called interactively via the GUI and using a graphical and menu driven interface.

### *8.1.1 DASD Utilities*

The following table shows the utilities that are used for DASD image file maintenance:

| Utility Name | Function |
|---|---|
| CCKDCDSK | CCKD DASD file integrity verification, recovery and repair utility |
| CCKDCOMP | CCKD DASD file compression utility |
| CCKDDIAG | CCKD DASD file diagnostics utility |
| CKDSWAP | CCKD DASD file swap-endian program |
| CKD2CCKD | Copy CKD DASD file to CCKD DASD file |
| CCKD2CKD | Copy CCKD DASD file to CKD DASD file |
| DASDCAT | Display PDS datasets and members |
| DASDCONV | DASD image file conversion program |
| DASDCOPY | Copy DASD file to another DASD file |
| DASDINIT | DASD image file creation |
| DASDISUP | Fix XCTL tables in SVCLIB |
| DASDLOAD | DASD loader program |
| DASDLS | List datasets on a volume |
| DASDPDSU | PDS unload utility |
| DASDSEQ | Display sequential datasets |

**Table 15: DASD image file maintenance utilites**

### 8.1.2 TAPE Utilities

The following table shows the utilities that are used for TAPE file maintenance:

| Utility Name | Function |
|---|---|
| HETGET | Extract files from an AWS or HET tape file |
| HETINIT | Initialize an AWS or HET tape file |
| HETMAP | Show information about a HET or AWS tape file |
| HETUPD | Update and/or copy an AWS or HET tape file |
| TAPECOPY | Copy a SCSI tape to or from an AWSTAPE disk file |
| TAPEMAP | Show information about an AWS tape file |
| TAPESPLT | Split an AWS tape file |

**Table 16: TAPE file maintenance utilities**

### 8.1.3 Miscellaneous Utilities

The following table shows utilities used for miscellaneous functions:

| Utility Name | Function |
|---|---|
| DMAP2HRC | P/390 DEVMAP conversion program |

**Table 17: Miscellaneous utilities**

### 8.1.4 Third Party Utilities

The following table shows “third party” utilities used for various functions:

| Utility Name | Function |
|---|---|
| PRTPUB | Mainframe print publishing program |

**Table 18: Third party utilities**

### *8.1.5 Mainframe Utilities*

The utilities according to the next table will run on the mainframe and provide various functionality that is useful for Hercules users:

| Utility Name | Function |
|---|---|
| AWSSL | AWS virtual tape utility |
| AWSUTIL | AWS format tape file generation utility |
| RAWSTAPE | Reverse AWSTAPE utility |
| VTTS | Virtual tape transportation system |

**Table 19: Mainframe utilities**

## 8.2 CCKDCDSK (CCKD DASD file integrity verification, recovery and repair utility)

### *8.2.1 Function*

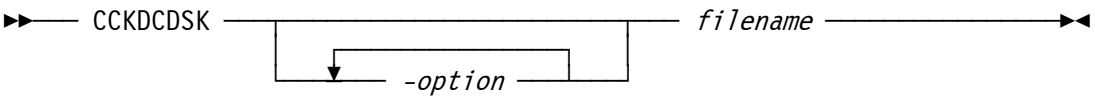
The CCKDCDSK utility performs compressed or shadowed CKD DASD emulation file integrity verification and recovery and repair. Calling the utility without any arguments will display a help information.

### *8.2.2 Syntax*

**Descriptive**

CCKDCDSK [*-option* [*-option* ... ]] *filename*

**Diagram**




### *8.2.3 Parameter*

***Options:***

**-v** Display version information and exit.

**-f** Force check, even if OPENED bit is on.

**-ro** Open the DASD file read-only, no repairs are done.

**-level** The level of checking that is to be performed. The higher the level the longer the integrity check takes. The level is any number from 1 - 4.

- Level 1 performs a minimal check. Device headers are verified, free space is verified, primary lookup table and secondary lookup tables are verified.

- Level 2 performs a normal check. These are the same checks as in level 1 plus all 5-byte track headers are verified.

- Level 3 performs the maximum checking. These are the same checks as in level 1 plus all track images are read, uncompressed and verified.

- Level 4 recovers everything.

***Arguments:***

**filename** Name of the CCKD DASD image file to be checked.

## 8.2.4 Examples

**Example 1:**

Perform a minimal checking of a CCKD DASD file without doing any repairs.

```
CCKDCDSK -0 –ro D:\MVS\DASD\MVSRES.CCKD
```

**Example 2:**

Perform a maximum level check and repair of a CCKD DASD file and force the check even the OPENED bit is set on.

```
CCKDCDSK -r –f D:\MVS\DASD\MVSRES.CCKD
```

## 8.2.5 Sample Utility Output

**Sample 1:**

Perform a maximum level check and repair of a CCKD DASD file and force the check even the OPENED bit is set on.

```
D:\HERCULES>cckdcdsk -3 -ro D:\MVS\DASD\MVSRES.CCKD
size 301956072 used 301956072 free 0 imbed 0 first 0x0 number 0
```

**Figure 201: CCKDCDSK utility output**

## 8.3 CCKDCOMP (CCKD DASD file compression utility)

### *8.3.1 Function*

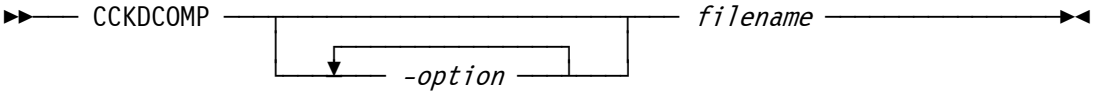
The CCKDCOMP utility removes all free space from a compressed or shadow CKD DASD emulation file. If *level* is specified then the CCKDCDSK utility is called first with the specified level; this is a short-hand method to call both functions in one utility call.

### *8.3.2 Syntax*

**Descriptive**

CCKDCOMP [*-option* [*-option* ... ]] *filename*

**Diagram**




### *8.3.3 Parameter*

***Options:***

**-v** Display version information and exit.

**-f** Force check even if OPENED bit is on.

**-level** The level of checking that is to be performed. The higher the level the longer the integrity check takes. The level is any number from 1 - 4.

- Level 1 performs a minimal check. Device headers are verified, free space is verified, primary lookup table and secondary lookup tables are verified.

- Level 2 performs a normal check. These are the same checks as in level 1 plus all 5-byte track headers are verified.

- Level 3 performs the maximum checking. These are the same checks as in level 1 plus all track images are read, uncompressed and verified.

- Level 4 recovers everything.

***Arguments:***

**filename** Name of DASD image file to be compressed.

### 8.3.4 Examples

**Example 1:**

Compress a CCKD DASD file.

```
CCKDCOMP D:\MVS\DASD\MVSRES.CCKD
```

**Example 2:**

Compress a CCKD DASD file and perform integrity checking and repair.

```
CCKDCOMP -3 D:\MVS\DASD\MVSRES.CCKD
```

### 8.3.5 Sample Utility Output

**Sample 1:**

Compress a CCKD DASD file and perform integrity checking and repair.

```
D:\HERCULES>cckdcomp -3 D:\MVS\DASD\MVSRES.CCKD
cckdcomp: completed: moves 33665  moved 301552856  freed 407400
```

**Figure 202: CCKDCOMP utility output**

## 8.4 CCKDDIAG (CCKD DASD file diagnostics utility)

### *8.4.1 Function*

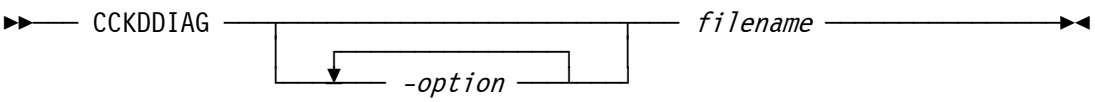
The CCKDDIAG utility is the diagnostic program which allows you to examine various aspects of a CCKD DASD image file.

### *8.4.2 Syntax*

**Descriptive**

CCKDDIAG [*-option* [*-option* ... ]] *filename*

**Diagram**




### *8.4.3 Parameter*

***Options:***

**-v** Display version information and exit.

**-d** Display DEVHDR.

**-c** Display CDEVHDR.

**-1** Display L1TAB (numeric one).

**-g** Enable debug output.

***CKD track related options:***

**-a *cc hh*** Display absolute *CCHH* data.

**-r *tt*** Display relative *TT* data.

**-2** Display L2TAB related to *-a* or *-r*.

**-t** Display track data.

**-x** Hex display track / key data.

***Offset option:***

**-o *oo ll*** Hex display data at offset *oo* of length *ll*.

***Arguments:***

**filename** Name of DASD image file to be compressed.

## *8.4.4 Examples*

**Example 1:**

Display the DEVHDR and CDEVHDR for a CCKD DASD image file.

```
CCKDDIAG -d -c D:\MVS\DASD\MVSRES.CCKD
```

**Example 2:**

Display the L1TAB and enable debug output for a CCKD DASD image file.

```
CCKDDIAG -1 -g D:\MVS\DASD\MVSRES.CCKD
```

**Example 3:**

Display relative TT data for a CCKD DASD image file.

```
CCKDDIAG -r 15 D:\MVS\DASD\MVSRES.CCKD
```

## *8.4.5 Sample Utility Output*

**Sample 1:**

Display the DEVHDR and CDEVHDR for a CCKD DASD image file and additionally enable debug output.

```
D:\Hercules>cckddiag -d -c -g D:\MVS\DASD\MVSRES.CCKD

READPOS seeking 0 (0x00000000)
READPOS reading buf addr 0012FA68 length 512 (0x00000200)

DEVHDR - 512 (decimal) bytes:
+0000 434B445F 43333730 1E000000 004C0000  C.D^C370.....<..
+0010 50000000 00000000 00000000 00000000  &...............
+0020 00000000 00000000 00000000 00000000  ................

3350 device has 30 heads/cylinder

READPOS seeking 512 (0x00000200)
READPOS reading buf addr 0012FC68 length 512 (0x00000200)

CDEVHDR - 512 (decimal) bytes:
+0000 00030141 42000000 00010000 0CFF5901  ...AB.........Y.
+0010 21F74901 19631500 EB071000 5CE90E00  !7I..c......*Z..
+0020 0B000000 00000000 30020000 00010500  ........0.......
+0030 00000000 00000000 00000000 00000000  ................
```

```
MAKBUF malloc L1TAB buffer of 264 bytes at 003B2860

READPOS seeking 1024 (0x00000400)
READPOS reading buf addr 003B2860 length 264 (0x00000108)
```

**Figure 203: CCKDDIAG utility output**

## 8.5 CCKDSWAP (CCKD DASD file swap-endian program)

### *8.5.1 Function*

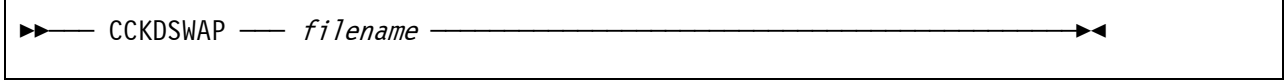
The CCKDSWAP utility is the swap-endian program, it swaps the byte order of a CCKD file.

### *8.5.2 Syntax*

**Descriptive**

CCKDSWAP *filename*

**Diagram**




### *8.5.3 Parameter*

**filename** Name of CCKD DASD image file to have its byte order swapped.

### *8.5.4 Examples*

**Example 1:**

Swap byte order for a CCKD DASD image file.

```
CCKDSWAP D:\MVS\DASD\MVSRES.CCKD
```

### *8.5.5 Sample Utility Output*

**Sample 1:**

Swap byte order for a CCKD DASD image file.

```
D:\HERCULES>cckdswap D:\MVS\DASD\MVSRES.CCKD
Hercules cckd swap-endian program Version 3.05
(c)Copyright 1999-2005 by Roger Bowler, Jan Jaeger, and others
cckdswap: D:\MVS\DASD\MVSRES.CCKD changed from little-endian to big-endian
```

**Figure 204: CCKDSWAP utility output**

## 8.6 CKD2CCKD (Copy CKD DASD file to CCKD DASD file)

### *8.6.1 Function*

The CKD2CCKD is used to copy (convert) an uncompressed CKD DASD file to a compressed CCKD DASD file. This utility is obsolete, it is replaced through the DASDCOPY utility which performs the same and additional functions. For details see DASDCOPY utility.

### *8.6.2 Syntax*

See DASDCOPY utility.

### *8.6.3 Parameter*

See DASDCOPY utility.

### *8.6.4 Examples*

See DASDCOPY utility.

### *8.6.5 Sample Utility Output*

See DASDCOPY utility.

## 8.7 CCKD2CKD (Copy CCKD DASD file to CKD DASD file)

### *8.7.1 Function*

The CCKD2CKD is used to copy (convert) a compressed CCKD DASD file to an uncompressed CKD DASD file. This utility is obsolete, it is replaced through the DASDCOPY utility which performs the same and additional functions. For details see DASDCOPY utility.

### *8.7.2 Syntax*

See DASDCOPY utility.

### *8.7.3 Parameter*

See DASDCOPY utility.

### *8.7.4 Examples*

See DASDCOPY utility.

### *8.7.5 Sample Utility Output*

See DASDCOPY utility.

## 8.8 DASDCAT (Display PDS datasets and members)

### *8.8.1 Function*

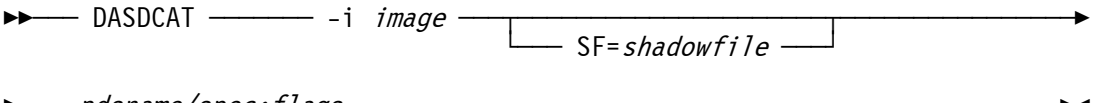
The DASDCAT utility displays pds datasets and members from a DASD image file. It can generate a list of members in a given PDS dataset or can list the contents of a single member or all members in a PDS.

### *8.8.2 Syntax*

**Descriptive**

DASDCAT -i *image* [SF=*shadowfile*] *pdsname/spec:flags*

**Diagram**




### *8.8.3 Parameter*

**-i image** The name (and optionally path) of the DASD image file.

**shadowfile** The (optional) name of an associated shadow DASD file.

**pdsname** The name of the partitioned dataset from which information will be extracted.

**spec** *spec* can be one of the following:

- The name of a PDS member to be listed.
- An asterisk (“*”), meaning that all members in the PDS are to be listed.
- A question mark (“?”) which causes the list of members in the PDS to be returned.

**flags** *flags* can be one of the following:

- “c” (list the member(s) as card images)
- “a” (list the member(s) in ASCII)

If no arguments are given the program displays help text and exits.

## 8.8.4 Examples

**Example 1:**

Create a member list of PDS SYS1.PARMLIB on volume MVSRES.CCKD.

```
DASDCAT -i D:/MVS/DASD/MVSRES.CCKD SYS1.PARMLIB/?
```

**Example 2:**

Print member IEASLP00 from PDS SYS1.PARMLIB on volume MVSRES.CCKD as card images.

```
DASDCAT -i D:/MVS/DASD/MVSRES.CCKD SYS1.PARMLIB/IEASLP00:c
```

**Example 3:**

Print all members from PDS SYS1.PARMLIB on volume MVSRES.CCKD as card images.

```
DASDCAT -i D:/MVS/DASD/MVSRES.CCKD SYS1.PARMLIB/*:c
```

## 8.8.5 Sample Utility Output

**Sample 1:**

Create a member list of PDS SYS1.PARMLIB on volume MVSRES.CCKD

```
D:\Hercules>dasdcat -i D:/MVS/DASD/MVSRES.CCKD SYS1.PARMLIB/?
Hercules DASD cat program Version 3.05
(c)Copyright 1999-2006 by Roger Bowler, Jan Jaeger, and others
commnd00
gtfparm
ieaabd00
ieaapf00
ieabld00
ieadmp00
.
.
.
lnklst00
mvikey00
rpfkey00
smfprm00
tsokey00
vatlst00
```

**Figure 205: DASDCAT utility output (create PDS member list)**

**Sample 2:**

Print member IEAIPS00 from PDS SYS1.PARMLIB on volume MVSRES.CCKD as card images.

```
D:\Hercules>dasdcat -i D:/MVS/DASD/MVSRES.CCKD SYS1.PARMLIB/IEAIPS00:c
Hercules DASD cat program Version 3.05
(c)Copyright 1999-2006 by Roger Bowler, Jan Jaeger, and others
CPU=10.0,IOC=5.0,MSO=3.0              /* DEFAULT IPS --IEAIPS00--   */
WKL=(1,50,99,100)
OBJ=1,SRV=(1000,*,*,0)                /* FIRST PERIOD BATCH  (DMN1) */
OBJ=2,SRV=(1000,*,0)                  /* SECOND PERIOD BATCH (DMN1) */
OBJ=3,SRV=(1000,*,*,0)                /* FIRST PERIOD TSO    (DMN2) */
OBJ=4,SRV=(1000,*,*,0)                /* SECOND PERIOD TSO   (DMN2) */
OBJ=5,SRV=(1000,*,*,0)                /* THIRD PERIOD TSO    (DMN3) */
OBJ=6,SRV=(1000,1000,1000,1000)       /* HOT BATCH           (DMN1) */
OBJ=7,SRV=(1000,1,0)                  /* LOW PRIORITY BATCH  (DMN1) */
DMN=1,CNSTR=(1,50,1)                  /* BATCH                      */
DMN=2,CNSTR=(1,50,255)                /* SHORT AND MEDIUM TSO       */
DMN=3,CNSTR=(1,50,16)                 /* LONG TSO                   */
PGN=1,(DMN=1,APG=6,DUR=10K,ISV=10K,OBJ=1,RTB=0)     /* BATCH -SHORT  */
      (DMN=1,APG=6,ISV=10K,OBJ=2,RTB=0)             /* -------MEDIUM */
PGN=2,(DMN=2,APG=14,DUR=600,ISV=600,OBJ=3,RTB=0)    /* TSO   -SHORT  */
      (DMN=2,APG=13,DUR=1400,ISV=2K,OBJ=4,RTB=0)    /* -------MEDIUM */
      (DMN=3,APG=6,ISV=10K,OBJ=5,RTB=0)             /* -------LONG   */
PGN=3,(DMN=1,APG=9,ISV=100K,OBJ=6,RTB=0)            /* HOT BATCH     */
PGN=4,(DMN=1,APG=6,ISV=10K,OBJ=7,RTB=0)             /* LOW PRTY BATCH*/
```

**Figure 206: DASDCAT utility output (list a member from a PDS)**

**Sample 3:**

Print all members from PDS SYS1.PARMLIB on volume MVSRES.CCKD as card images.

```
D:\Hercules>dasdcat -i D:/MVS/DASD/MVSRES.CCKD SYS1.PARMLIB/*:c
Hercules DASD cat program Version 3.05
(c)Copyright 1999-2006 by Roger Bowler, Jan Jaeger, and others
> Member COMMND00
| TOD=NOPROMPT
> Member GTFPARM
| TRACE=SYS,USR,TRC,DSP,PCI,SRM,RR,EXT,PI,SVC,IO,SIO
> Member IEAABD00
| SDATA=(ALLSDATA),PDATA=(ALLPDATA)
> Member IEAAPF00
|  SYS1.LINKLIB                      MVSRES,
|  SYS1.SVCLIB                       MVSRES,
|  SYS1.LPALIB                       MVSRES,
|  SYS1.VTAMLIB                      MVSRES,
|  SYS1.INDMAC                       MVSRES
> Member IEABLD00
| SYS1.LINKLIB ALLOC,ALLOCATE,E,EDIT,HEWL,
|                IEWL,IEWLF128,IEWLF440,IEWLF880,IFOX01,IFOX02,IFOX03,
|                IFOX04,IFOX05,IFOX06,IFOX11,IFOX21,IFOX31,IFOX41,IFOX51,
|                IFOX61,IFOX62,LINK,LINKEDIT,LOADER,
|                LOGOFF,LOGON,SUBMIT,TEST
.
.
.
several lines not displayed
.
.
.
> Member TSOKEY00
| USERMAX=8,                       /* MAXIMUM TSO USERS                */
| RECONLIM=60,                     /* MAXIMUM DISCONNECT MINUTES       */
| BUFRSIZE=132,                    /* VTIOC BUFFER SIZE                */
| HIBFREXT=48000,                  /* MAX BUFFERS BEFORE SWAP OUT      */
| LOBFREXT=24000,                  /* MIMIMUM BUFFERS BEFORE SWAP IN   */
| MODE=NOBREAK,                    /* KEYBOARD LOCK OPTION             */
| MODESW=NO,                       /* MODESWITCH FROM TERMINAL OPTION  */
| CHNLEN=4,                        /* NO. OF RU'S PER CHAIN            */
| SCRSIZE=1920                     /* MAXIMUM SCREEN SIZE              */
```

**Figure 207: DASDCAT utility output (list all members of a PDS)**

## 8.9 DASDCONV (DASD image file conversion program)

### *8.9.1 Function*

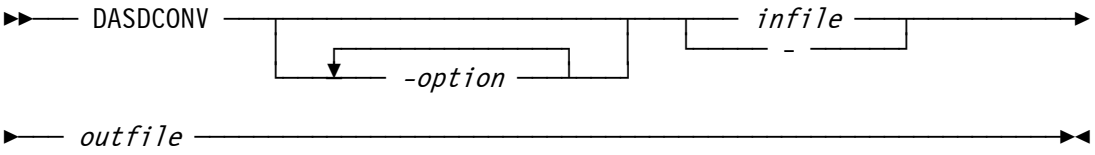
The DASD CKD image conversion utility is used to convert HDR-30 CKD image files or compressed (.gz) image files to the AWSCKD format that is used by the Hercules Emulator. The .gz compressed format was recently used by IBM to ship the ADCD DASD image files for the FLEX-ES emulator.

### *8.9.2 Syntax*

**Descriptive**

```
DASDCONV [-option [-option ... ]] {infile | - } outfile
```

**Diagram**




### *8.9.3 Parameter*

***Options:***

**-r** Replace the output file if it already exists.

**-lfs** Output CKD file will be a single file even if it exceeds 2 GB in size.

**-q** The "-q" (quiet) option suppresses the progress messages.

***Arguments:***

**infile** The name (and optionally path) of the input HDR-30 CKD image file or the name of the compressed (.gz) image file.

**-** If the hyphen is specified instead of a filename, then DASDCONV reads from stdin. This gives the possibility of piping the output from gunzip in the case that DASDCONV was generated without gzip support.

**outfile** The name of the AWSCKD image file to be created.

## 8.9.4 Examples

**Example 1:**

Convert a .gz compressed DASD image file to AWSCKD format. Create a large file (> 2 GB) and replace the output file if it already exists.

```
DASDCONV -r -lfs D:\MVS\DASD\MVSRES.GZ D:\MVS\DASD\MVSRES.CKD
```

## 8.9.5 Sample Utility Output

**Sample 1:**

Convert a .gz compressed DASD image file to AWSCKD format. Create a large file (> 2 GB) and replace the output file if it already exists.

```
D:\HERCULES>DASDCONV -r -lfs D:\MVS\DASD\MVSRES.GZ D:\MVS\DASD\MVSRES.CKD
Hercules DASD CKD image conversion program
Version 3.05
(c)Copyright 1999-2005 by Roger Bowler, Jan Jaeger, and others
Converting 3390 volume MVSRES: 3339 cyls, 15 trks/cyl, 56832 bytes/track
3339 cylinders successfully written to file MVSRES.ckd
DASD conversion successfully completed.
```

**Figure 208: DASDCONV utility output**

## 8.10 DASDCOPY (Copy DASD file to another DASD file)

### *8.10.1 Function*

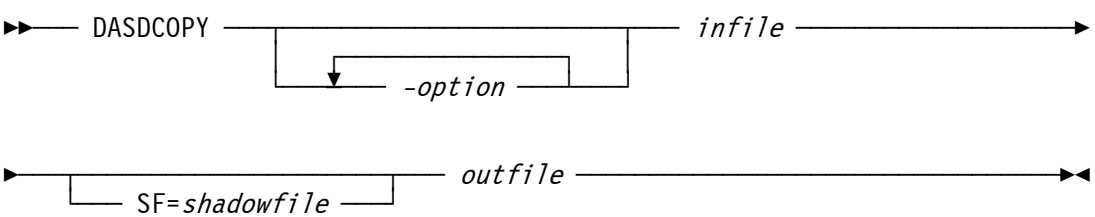
The DASDCOPY utility is used to copy a DASD image file to another DASD image file.

### *8.10.2 Syntax*

**Descriptive**

DASDCOPY [*-option* [*-option* ... ]] *infile* [SF=*shadowfile*] *outfile*

**Diagram**




### *8.10.3 Parameter*

***Options:***

| | |
|---|---|
| **-v** | Display version information and help text. |
| **-h** | Display usage summary and quit. |
| **-q** | Quiet mode, do not display status. |
| **-r** | Replace the output file if it already exists. |
| **-z** | Compress output using zlib (default). |
| **-bz2** | Compress output using bzip2. |
| **-0** | Do not compress output (0 = zero). |
| **-blks *n*** | Size of output FBA file. |
| **-cyls *n*** | Size of output CKD file. |
| **-a** | Output CKD file will have alternate cylinders. |
| **-lfs** | Output CKD file will be a single file even if it exceeds 2 GB in size. |
| **-o *type*** | Output file type (CKD, CCKD, FBA, CFBA). |

***Arguments:***

**infile** The input DASD file.

**shadowfile** The (optional) name of an associated shadow DASD file.

**outfile** The output DASD file.

## *8.10.4 Examples*

**Example 1:**

Copy the compressed DASD file (CCKD) MVSRES.CCKD to SARESX.148, uncompress it (CKD file) and replace an already existing file with the same name.

```
DASDCOPY -r -lfs D:/MVS/DASD/MVSRES.CCKD D:/MVS/DASD/MVSRESX.148
```

**Example 2:**

Copy the compressed DASD file (CCKD) MVSRES.CCKD to SARESX.148, uncompress it (CKD file) and create 2 separate files for systems which do not support large file support.

```
DASDCOPY D:/MVS/DASD/MVSRES.CCKD D:/MVS/DASD/MVSRESX.148
```

## *8.10.5 Sample Utility Output*

**Sample 1:**

Copy the DASD image file MVSRES.CCKD to the new DASD image file SARESX.148. Do not create a large image file (> 2 GB), create two separate files instead.

```
D:\HERCULES>dasdcopy D:/MVS/DASD/MVSRES.CCKD D:/MVS/DASD/SARESX.148
HHCDU044I Creating 3390 volume : 3339 cyls, 15 trks/cyl, 56832 bytes/track
HHCDU041I 2519 cylinders successfully written to file D:/MVS/DASD/SARESX_1.148
HHCDU041I 820 cylinders successfully written to file D:/MVS/DASD/SARESX_2.148
/ 100%   50085 of 50085
HHCDC010I Copy successful !!!
```

**Figure 209: DASDCOPY utility output**

## 8.11 DASDINIT (DASD image file creation)

### *8.11.1 Function*

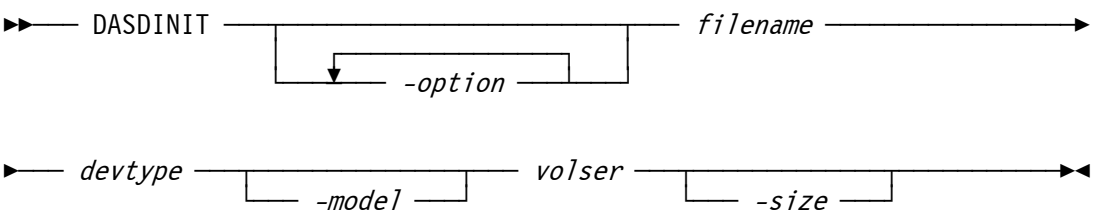
The Hercules DASD image file creation program (DASDINIT) is used to create an empty DASD image file.

### *8.11.2 Syntax*

**Descriptive**

DASDINIT [*-option* [*-option* ... ]] *filename devtype*[*-model*] *volser* [*size*]

**Diagram**




### *8.11.3 Parameter*

***Options:***

**-v** Display version information and help text.

**-z** Build compressed DASD image file using zlib.

**-bz2** Build compressed DASD image file using bzip2.

**-0** Build compressed DASD image file with no compression (0 = zero).

**-lfs** Build a large (uncompressed) DASD file (single file), if it is supported on the platform DASDINIT is running on.

**-a** Build a DASD image file that includes alternate cylinders. This option is ignored if the size is manually specified.

**-r** Build raw DASD image file (no VOL1 or IPL track).

**-linux** Null track images will look like Linux DASDFMT'ed images (3390 device type only).

***Arguments:***

**filename** Name of DASD image file to be created.

**devtype** FBA: 0671, 3310, 3370, 9313, 9332, 9335, 9336
CKD: 2311, 2314, 3330, 3340, 3350, 3375, 3380, 3390, 9345
A complete list of all Hercules supported DASD device types and models can be found in Appendix A.

**model** Device model (Note – this implies size). A list of Hercules supported device types and models can be found in Appendix A.

**volser** Volume serial number (1-6 characters).

**size** Number of CKD cylinders or 512-byte FBA sectors (required if model not specified, otherwise optional).

For CKD volumes which exceed 2GB, such as the 3390-3, if the *-lfs* parameter is not specified the DASDINIT program will create multiple files by appending the characters _1, _2, _3 etc. to the file name specified on the command line. These characters are inserted before the first dot (**.**) occurring after the last slash (***/***). If there is no dot then the characters are appended to the end of the name. Each file contains a whole number of cylinders. Hercules CKD support recognizes the files as belonging to a single logical volume. Specify the full name of just the first file in the Hercules configuration file (e.g. "filename_1").

The DASDINIT program cannot create FBA volumes exceeding 2GB unless the *-lfs* parameter is specified and large file size is supported on your platform.

## *8.11.4 Examples*

**Example 1:**

Create a 3390 Model 3 with volume serial USR001, the file(s) should be named USR001.300 and place them into directory D:\MVS\DASD\. Note that the utility creates actually two files called USR001_1.300 and USR001_2.300 because no large file support is requested.

```
DASDINIT D:\MVS\DASD\USR001.300 3390-3 USR001
```

**Example 2:**

Create the same DASD volume as in the previous example but create just one file (large file suport) named USR001.300 in the same directory.

```
DASDINIT -lfs D:\MVS\DASD\USR001.300 3390-3 USR001
```

**Example 3:**

Create a 3390 Model 3 with volume serial SYS3A0 and alternate cylinder(s), the file should be compressed with ZLIB and the filename has to be TSTVOL.DSD. The DASD volume must be placed in directory D:\MVS\DASD\.

```
DASDINIT -z -a D:\MVS\DASD\TSTVOL.DSD 3390-3 SYS3A0
```

### *8.11.5 Sample Utility Output*

**Sample 1:**

Create a 3390 model 3 DASD image file named SYS3A0.3A0 with volser=SYS3A0.

```
D:\HERCULES>DASDINIT D:\MVS\DASD\SYS3A0.3A0 3390-3 SYS3A0
HHCDU044I Creating 3390 volume SYS3A0: 3339 cyls, 15 trks/cyl, 56832 bytes/track
HHCDU041I 2519 cylinders successfully written to file D:/MVS/DASD/SYS3A0_1.3A0
HHCDU041I 820 cylinders successfully written to file D:/MVS/DASD/SYS3A0_2.3A0
HHCDI001I DASD initialization successfully completed.
```

**Figure 210: DASDINIT utility output**

## 8.12 DASDISUP (Fix XCTL tables in SVCLIB)

### *8.12.1 Function*

The DASDISUP utility fixes the XCTL tables in SVCLIB on OS/360 systems. On an OS/360 system the OPEN / CLOSE / EOV modules in SYS1.SVCLIB have XCTL tables embedded within them. These tables contain TTRs, pointing to other modules, and these TTRs need to be adjusted after loading SVCLIB to DASD.

OS/360 provides a program called IEHIOSUP to perform this function, however the issue is that you cannot run IEHIOSUP until you have the system up and running, and you cannot IPL the system until you have fixed the XCTL tables.

To circumvent this problem Hercules provides a utility program called DASDISUP which can be run from the Unix or Windows command line after running DASDLOAD and fixes the XCTL tables.

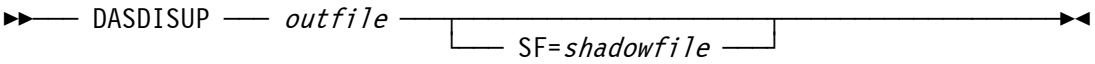
***Note:*** *Do not use this procedure except on OS/360 IPL volumes; other operating systems do not have XCTL tables.*

### *8.12.2 Syntax*

**Descriptive**

DASDISUP *outfile* [SF=*shadowfile*]

**Diagram**




### *8.12.3 Parameter*

**outfile** The name of the OS/360 IPL volume to be updated.

**shadowfile** The (optional) name of the associated shadow DASD file.

### *8.12.4 Examples*

**Example 1:**

Fix the XCTL tables in SVCLIB on volume OS36IP.148.

```
DASDISUP D:/MVS/DASD/OS36IP.148
```

### *8.12.5 Sample Utility Output*

**Sample 1:**

Fix the XCTL tables in SVCLIB on volume OS36IP.148.

```
D:\HERCULES>dasdisup D:/MVS/DASD/OS36IP.148
Hercules IEHIOSUP program Version 3.05
(c)Copyright 1999-2005 by Roger Bowler, Jan Jaeger, and others
HHCDS006W Member IGG019PZ is not single text record
HHCDS006W Member IGG019V6 is not single text record
HHCDS007W Member IGG019V6 size 0A20 exceeds X'7F8' bytes
HHCDS002I End of directory: 4 members selected
HHCDS010I Member IGG019PX skipped
HHCDS010I Member IGG019PY skipped
HHCDS010I Member IGG019PZ skipped
HHCDS010I Member IGG019V6 skipped
```

**Figure 211: DASDISUP utility output**

## 8.13 DASDLOAD (DASD loader program)

### *8.13.1 Function*

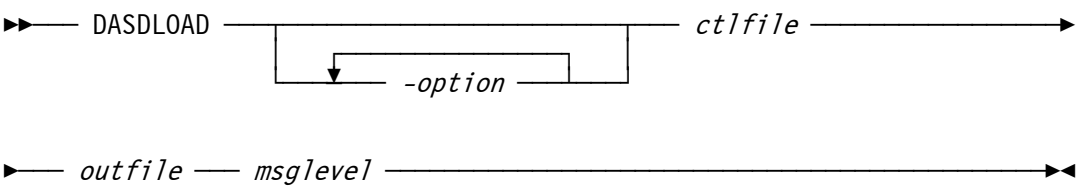
The Hercules DASD loader program is used to create new DASD file images and load them with data from unloaded PDS files (e.g. a list of TSO XMIT files or a sequential file).

### *8.13.2 Syntax*

**Descriptive**

DASDLOAD [*-option* [*-option* ... ]] *ctlfile outfile msglevel*

**Diagram**




### *8.13.3 Parameter*

***Options:***

| | |
|---|---|
| **-z** | Build compressed DASD image file using zlib. |
| **-bz2** | Build compressed DASD image file using bzip2. |
| **-0** | Build compressed DASD image file with no compression (0 = zero). |
| **-lfs** | Build a large DASD image file (can exceed 2 Gb in size). |
| **-a** | Build DASD image file that includes alternate cylinders. |

***Arguments:***

| | |
|---|---|
| **ctlfile** | This is the name of the control file that specifies the datasets to be loaded onto the newly created volume. |
| **outfile** | The name of the DASD image file to be created. |
| **msglevel** | Msglevel can be a number from 0 to 5, it controls the detail level of messages issued during the load. |

## *8.13.4 Control File*

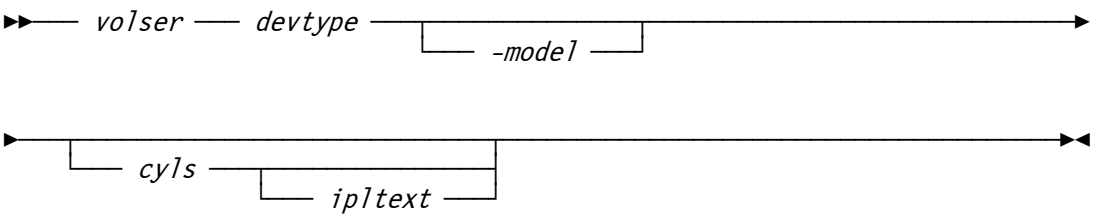
The control file required by the DASDLOAD program is an ASCII text file consisting of a *volume statement* followed by one *dataset statement* for each dataset to be created.

### 8.13.4.1 Volume Statement

**Descriptive**

*volser devtype*[*-model*] [*cyls* [*ipltext*]]

**Diagram**




where:

**volser** Volser is the volume serial number of the newly created volume.

**devtype** This specifies the emulated device type (2311, 3330, 3350, 3375, 3380, 3390) of the new volume. FBA device types are not supported by the DASDLOAD program. A list of Hercules supported device types and models can be found in Appendix A.

**model** This specifies the device model. The model implies the size of the volume. A list of Hercules supported device types and models can be found in Appendix A.

**cyls** Cyls specifies the size of the new volume in cylinders. If *cyls* is coded as "*" (asterisk), as "0" or is omitted, then the deafult size for the device type and model is used. Cylinders is ignored for compressed devices.

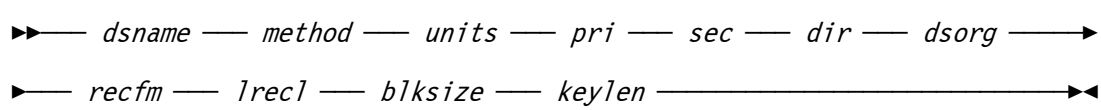
**ipltext** An optional parameter specifying the name of a file containing the IPL text which will be written to the volume. The file must be in the form of an object deck containing fixed length 80-byte records in the same format as expected by IBCDASDI or ICKDSF.

### 8.13.4.2 Dataset Statement

**Descriptive**

*dsname method units pri sec dir dsorg recfm lrecl blksize keylen*

**Diagram**




where:

**dsname** The name of the dataset to be created.

**method** The dataset loading method can be one of the following:

| | |
|---|---|
| **XMIT filename** | The dataset is loaded from an unloaded PDS created by the TSO XMIT command. The input is a binary file containing fixed length 80 byte records with no record delimiters. |
| **VS filename** | The dataset is loaded from an unloaded PDS created by IEBCOPY. The input is a binary file containing variable length spanned records with the record descriptor words omitted and with no record delimiters. |
| **SEQ filename** | The dataset is loaded from an binary file. ASCII / EBCDIC conversion is not currently supported. Also the DSORG must be PS or DA and RECFM must either be F or FB. |
| **XMSEQ filename** | The dataset is loaded from a dump of a sequential dataset created by the TSO XMIT command. The input is a binary file containing fixed length 80 byte records with no record delimiters. |
| **EMPTY** | The dataset is initialized with an end-of-file record (if DSORG is PS) or is an empty PDS directory (if DSORG is PO). |
| **DIP** | The dataset is initialized with a LOGREC header record. |
| **CVOL** | The dataset is initialized as an OS SYSCTLG containing the minimum entries needed to IPL an OS/360 system. |
| **VTOC filename** | Specifies the size and location of the VTOC. A dataset name must be coded on this statement although it is not used. If no VTOC statement is present, the VTOC will be placed after the last dataset on the volume and the size of the VTOC will be the minimum number of tracks necessary. |

**units** This specifies the space allocation units: **TRK** or **CYL**.

**pri** This is the space allocation primary quantity.

**sec** This is the space allocation secondary quantity.

**dir** This is the number of directory blocks.

**dsorg** This specifies the dataset organization: **PS**, **PO**, **DA** or **IS**.

**recfm** This is the record format: **F**, **FB**, **FBS**, **V**, **VB**, **VBS** or **U**.

**lrecl** This is the logical record length.

**blksize** This is the block size.

**keylen** This is the key length.

All parameters except the dsname and method are optional. Defaults of zero are supplied for DCB parameters. For datasets loaded with the XMIT method the DCB parameters are taken from the unloaded PDS, minimum space allocation required to load the dataset is used unless a larger quantity is specified. If space allocation is omitted the default is **TRK 1 0 0**. If CYL is specified without any primary quantity then the default allocation is 1 cylinder or the minimum number of cylinders required to load the dataset, whichever is larger.

## *8.13.5 Examples*

**Example 1:**

Create a 2314 volume in a file called SYSRES.230 using the control file SYSRES.PLF with message level 2.

```
DASDLOAD SYSRES.PLF SYSRES230 2
```

The corresponding control file SYSRES.PLF looks like the following:

```
# Pack layout file for MFT system residence volume
sysres 2314 * ieaipl00.rdr
sys1.parmlib    xmit    /cdrom/os360/reslibs/parmlib.xmi
sys1.imagelib   xmit    /cdrom/os360/reslibs/imagelib.xmi
sysctlg         cvol    trk 1 0 0    ps f 256 256 8
sysvtoc         vtoc    trk 5
sys1.logrec     dip     trk 1 0 0
sys1.nucleus    xmit    /cdrom/os360/reslibs/nucleus.xmi cyl
sys1.svclib     xmit    /cdrom/os360/reslibs/svclib.xmi cyl
sys1.sysjobqe   empty   cyl  2 0 0   da f 176 176 0
sys1.dump       empty   cyl 10 0 0   ps u 0 3625 0
```

**Example 2:**

Create a compressed 3390-3 volume in a file called LINUX.500 containing a bootable linux system for Linux/390 installation using the control file LINUX PRM with message level 5.

```
DASDLOAD -z LINUX.PRM LINUX.500 5
```

The corresponding control file LINUX.PRM looks like the following:

```
#
# Build a bootable linux disk
#   [Note: the dataset names (sys1.linux. ...) are hard-coded in
#          linuxipl.obj and cannot be changed without rebuilding it]
#
linux  3390-3 * linuxipl.obj
sys1.linux.parmfile    SEQ images/redhat.prm trk   1 0 0 ps fb 1024 1024
sys1.linux.tapeipl.ikr SEQ images/kernel.img trk 200 0 0 ps fb 1024 1024
sys1.linux.initrd      SEQ images/initrd.img trk 200 0 0 ps fb 1024 1024
```

### *8.13.6 Sample Utility Output*

**Sample 1:**

Create a 3390 model 3 in a compressed file called TST001.A80 containing 3 libraries (which have been downloaded from CBT tape as XMIT files) using the control file TST001.DAT with message level 1.

The control file TST001.DAT looks like follows:

```
TST001 3390-3 *
SYS2.RACF.UTIL    XMIT    D:\MVS\DASD\CBT728.XMI
SYS2.BATCH.CNTL   XMIT    D:\MVS\DASD\CBT357.XMI
SYS2.PL1.SAMPLES  XMIT    D:\MVS\DASD\CBT316.XMI
```

The corresponding output from the utility is the following:

```
D:\HERCULES>DASDLOAD -z D:\MVS\DASD\TST001.DAT D:\MVS\DASD\TST001.A80 1
Hercules DASD loader program Version 3.05
(c)Copyright 1999-2005 by Roger Bowler, Jan Jaeger, and others
--------- TST001 3390-3 *
HHCDL006I Creating 3390 volume TST001: 15 trks/cyl, 56832 bytes/track
HHCDU044I Creating 3390 volume TST001: 3339 cyls, 15 trks/cyl, 56832 bytes/track
HHCDU041I 3339 cylinders successfully written to file D:\MVS\DASD\TST001.A80
HHCDL009I Loading 3390 volume TST001
--------- SYS2.RACF.UTIL    XMIT    D:\MVS\DASD\CBT728.XMI
HHCDL012I Creating dataset SYS2.RACF.UTIL at cyl 0 head 1
HHCDL110I Processing file D:\MVS\DASD\CBT728.XMI
--------- SYS2.BATCH.CNTL   XMIT    D:\MVS\DASD\CBT357.XMI
HHCDL012I Creating dataset SYS2.BATCH.CNTL at cyl 0 head 2
HHCDL110I Processing file D:\MVS\DASD\CBT357.XMI
--------- SYS2.PL1.SAMPLES  XMIT    D:\MVS\DASD\CBT316.XMI
HHCDL012I Creating dataset SYS2.PL1.SAMPLES at cyl 10 head 0
HHCDL110I Processing file D:\MVS\DASD\CBT316.XMI
HHCDL057I VTOC starts at cyl 21 head 10 and is 1 track
HHCDL014I Free space starts at cyl 21 head 11
HHCDL016I Total of 3339 cylinders written to D:\MVS\DASD\TST001.A80
```

**Figure 212: DASDLOAD utility output (XMIT method, message level 1)**

**Sample 2:**

Create a 3390 model 3 in a compressed file called TST001.A80 containing a library contained in a XMIT files using the control file TST001.DAT with message level 3.

The control file TST001.DAT looks like follows:

```
TST001 3390-3 *
SYS2.RACF.UTIL    XMIT    D:\MVS\DASD\CBT728.XMI
```

The corresponding output from the utility is the following:

```
D:\HERCULES>DASDLOAD -z D:\MVS\DASD\TST001.DAT D:\MVS\DASD\TST001.A80 3
Hercules DASD loader program Version 3.05
(c)Copyright 1999-2005 by Roger Bowler, Jan Jaeger, and others
--------- TST001 3390-3 *
HHCDL006I Creating 3390 volume TST001: 15 trks/cyl, 56832 bytes/track
HHCDU044I Creating 3390 volume TST001: 3339 cyls, 15 trks/cyl, 56832 bytes/track
HHCDU041I 3339 cylinders successfully written to file D:\MVS\DASD\TST001.A80
HHCDL009I Loading 3390 volume TST001
--------- SYS2.RACF.UTIL XMIT D:\MVS\DASD\CBT728.XMI
HHCDL012I Creating dataset SYS2.RACF.UTIL at cyl 0 head 1
HHCDL110I Processing file D:\MVS\DASD\CBT728.XMI
HHCDL078I File 1: DSNAME=SBGOLOB.CBT470.FILE728
HHCDL079I DSORG=PO RECFM=FB LRECL=80 BLKSIZE=5600 KEYLEN=0 DIRBLKS=6
HHCDL084I Original dataset: DSORG=PO RECFM=FB LRECL=80 BLKSIZE=5600 KEYLEN=0
HHCDL085I Dataset was unloaded from device type 3030200F (3390)
HHCDL086I Original device has 2226 cyls and 15 heads
HHCDL095I Member $$$#DATE TTR=000214 Userdata=044600000105311F ........
  0105311F0733000C 000C0000C3C2E360 F4F7F0404040     ............CBT-470
HHCDL095I Member $$NOTE1  TTR=00020E Userdata=010100160105311F ........
  0105311F07330009 000A0000E2C2C7D6 D3D6C2404040     ............SBGOLOB
HHCDL095I Member @FILE728 TTR=000210 Userdata=044600000105311F ........
  0105311F07330005 00050000C3C2E360 F4F7F0404040     ............CBT-470
HHCDL095I Member ICHPWX01 TTR=000109 Userdata=010100000105311F ........
  0105311F0711019A 019A0000D1D6E4E2 D4C140404040     ............JOUSMA
HHCDL090I End of directory
HHCDL013I Dataset SYS2.RACF.UTIL contains 1 track
HHCDL057I VTOC starts at cyl 0 head 2 and is 1 track
HHCDL014I Free space starts at cyl 0 head 3
HHCDL016I Total of 3339 cylinders written to D:\MVS\DASD\TST001.A80
```

**Figure 213: DASDLOAD utility output (XMIT method, message level 3)**

**Sample 3:**

Create a 3390 model 3 in a compressed file called TST001.A80 containing a sequential file using the control file TST001.DAT with message level 5.

The control file TST001.DAT looks like follows:

```
TST001 3390-3 *
SYS2.SEQ.DATA SEQ D:\MVS\DASD\SEQ.BIN TRK 5 5 0 PS FB 80 3120
```

The corresponding output from the utility is the following:

```
D:\HERCULES>dasdload -z D:\MVS\DASD\TST001.DAT D:\MVS\DASD\TST001.A80 5
Hercules DASD loader program Version 3.05
(c)Copyright 1999-2005 by Roger Bowler, Jan Jaeger, and others
--------- TST001 3390-3 *
HHCDL006I Creating 3390 volume TST001: 15 trks/cyl, 56832 bytes/track
HHCDU044I Creating 3390 volume TST001: 3339 cyls, 15 trks/cyl, 56832 bytes/track
HHCDU041I 3339 cylinders successfully written to file D:\MVS\DASD\TST001.A80
HHCDL009I Loading 3390 volume TST001
--------- SYS2.SEQ.DATA SEQ D:\MVS\DASD\SEQ.BIN TRK 5 5 0 PS FB 80 3120
HHCDL012I Creating dataset SYS2.SEQ.DATA at cyl 0 head 1
HHCDL013I Dataset SYS2.SEQ.DATA contains 5 tracks
---------
HHCDL057I VTOC starts at cyl 0 head 6 and is 1 track
HHCDL058I Format 4 DSCB CCHHR=0000000601 (TTR=000001)
+0000 00000000 00000000 00000000 04040404  ................
+0010 04040404 04040404 04040404 04040404  ................
Line 0020 same as above
+0030 04040404 04040404 F4000000 0603002F  ........4....../
+0040 0D0B0000 00008001 00000D0B 000FE5A2  ..............Vs
+0050 00000030 0000322D 00000000 00000000  ...0..2-........
+0060 00000000 00000000 00000000 00000000  ................
+0070 00000000 00010000 00000600 00000600  ................
+0080 00000000 00000000 00000000 00000000  ................
+0090 00000000 00000000                    ........
HHCDL058I Format 5 DSCB CCHHR=0000000602 (TTR=000002)
+0000 00000000 00000000 00000000 05050505  ................
+0010 00000000 00000000 00000000 00000000  ................
Line 0020 same as above
+0030 00000000 00000000 F5000000 00000000  ........5.......
+0040 00000000 00000000 00000000 00000000  ................
Lines 0050 to 0080 same as above
+0090 00000000 00000000                    ........
HHCDL058I Format 1 DSCB CCHHR=0000000603 (TTR=000003) SYS2.SEQ.DATA
+0000 00000000 00000000 00000000 E2E8E2F2  ............SYS2
+0010 4BE2C5D8 4BC4C1E3 C1404040 40404040  .SEQ.DATA
+0020 40404040 40404040 40404040 40404040
+0030 40404040 40404040 F1E3E2E3 F0F0F100          1TST001.
+0040 016A002C 00000001 4000C8C5 D9C3E4D3  .|.,.... .HERCUL
```

```
+0050 C5E24040 40404000 00000000 00004000  ES     ....... .
+0060 90000C30 00500000 00A08000 00050000  ...0.&..........
+0070 02DCDE00 00010000 00000100 00000500  ................
+0080 00000000 00000000 00000000 00000000  ................
+0090 00000000 00000000                    ........
HHCDL059I Format 0 DSCB CCHHR=0000000604 (TTR=000004)
HHCDL059I Format 0 DSCB CCHHR=0000000605 (TTR=000005)
.
.
.
several lines not displayed
.
.
.
HHCDL059I Format 0 DSCB CCHHR=0000000631 (TTR=000031)
HHCDL059I Format 0 DSCB CCHHR=0000000632 (TTR=000032)
HHCDL014I Free space starts at cyl 0 head 7
HHCDL016I Total of 3339 cylinders written to D:\MVS\DASD\TST001.A80
HHCDL017I Updating VTOC pointer 0000000601
```

**Figure 214: DASDLOAD utility output (SEQ method, message level 5)**

## 8.14 DASDLS (List datasets on a volume)

### *8.14.1 Function*

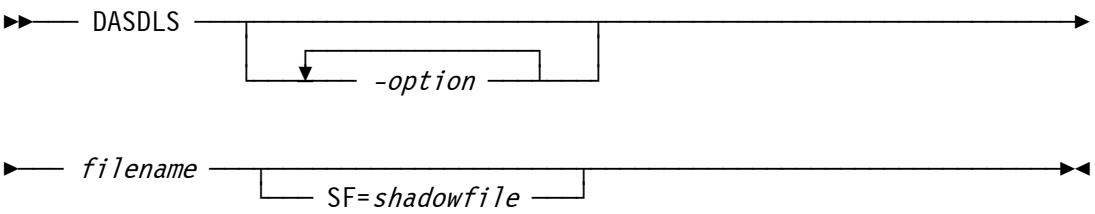
The DASDLS utility is used to list all datasets on the specified DASD image file.

### *8.14.2 Syntax*

**Descriptive**

DASDLS [*-option* [*-option* ... ]] *filename* [SF=*shadowfile*]

**Diagram**




### *8.14.3 Parameter*

***Options:***

**-info** Show Format 1 DSCB information. This option is implied if any of the other options are set.

**-caldt** Display dates as YYYYMMMDD. If this option is not specified then dates are displayed as YYDDD.

**-refdt** Display the last-referenced date (not applicable to MVT), otherwise it is omitted.

**-expdt** Display the expiry date, otherwise it is omitted.

**-hdr** Display column headers, otherwise they are omitted.

**-dsnl[=*n*]** Reserve space on the output line for dataset names up to '*n*' characters. If '*n*' is not specified then space for 26 characters is reserved. If the –dsnl option is omitted, then space for 44 characters is reserved.

**yroffs[=*n*]** Add the year offset '*n*' (which may be negative) to dates before displaying them. If '*n*' is omitted then 28 is used. No checking is done that the value of '*n*' is sensible.

***Arguments:***

**filename** Name of the DASD image file for which the datasets contained will be listed.

**shadowfile** Optional name of an associated shadow file (s) from which any datasets will be listed.

## 8.14.4 Examples

**Example 1:**

List all the datasets currently on the volume D:\MVS\DASD\MVSRES.CCKD.

```
DASDLS D:\MVS\DASD\MVSRES.CCKD
```

**Example 2:**

List all the datasets currently on the volume D:\MVS\DASD\MVSRES.CCKD showing information from the Format 1 DSCB, with column headers and reserving only 20 characters for the dataset name.

```
DASDLS –dsnl=20 –hdr D:\MVS\DASD\MVSRES.CCKD
```

**Example 3:**

List all the datasets currently on the volume D:\MVS\DASD\MVSRES.CCKD showing information from the Format1 DSCB, with column headers, calendar date format and reserving only 20 characters for the dataset name.

```
DASDLS –dsnl=20 –hdr –caldt D:\MVS\DASD\MVSRES.CCKD
```

## 8.14.5 Sample Utility Output

**Sample 1:**

List all the datasets currently on the volume D:\MVS\DASD\MVSRES.CCKD.

```
D:\Hercules>dasdls D:\MVS\DASD\ MVSRES.CCKD
Hercules DASD list program Version 3.10.0
(c)Copyright 1999-2010 by Roger Bowler, Jan Jaeger, and others
D:\MVS\DASD\MVSRES.CCKD: VOLSER=MVSRES
SYS1.STAGE1.OUTPUT
SYS1.LPALIB
SYS1.LINKLIB
SYS1.SVCLIB
SYS1.NUCLEUS
SYS1.DCMLIB
.
several lines not displayed
.
SYS1.SMPCRQ
SYS1.SMPSCDS
SYS1.COBLIB
SYS1.FORTLIB
SYS1.PL1LIB
SYS1.SORTLIB
```

**Figure 215: DASDLS utility output (basic information)**

**Sample 2:**

List all the datasets currently on the volume D:\MVS\DASD\MVSRES.CCKD showing information from the Format1 DSCB, with column headers, expiration date, calendar date format with a year offset of 10 years and reserving only 20 characters for the dataset name.

```
D:\Hercules>dasdls -caldt -expdt -hdr -dsnl=20 -yroffs=10 D:\MVS\DASD\MVSRES.CCKD
Hercules DASD list program Version 3.10.0
(c)Copyright 1999-2010 by Roger Bowler, Jan Jaeger, and others
D:\MVS\DASD\MVSRES.CCKD: VOLSER=MVSRES
Dsname                Created  Exp. Date ORG RECFM LRECL BLKSZ Key  Trks%Use#Ext 2ndry_alloc
SYS1.STAGE1.OUTPUT   1983Apr28 --------- PS  FB       80 19040   0    52  98   2 TRK       30
SYS1.LPALIB          1983Apr28 1981Dec16 PO  U           19069   0   900  50   1 CYL        1
SYS1.LINKLIB         1983Apr28 1981Dec16 PO  U           19069   0  1200  38   1 CYL        1
SYS1.SVCLIB          1983Apr28 1981Dec16 PO  U           19069   0    60  24   1 CYL        1
SYS1.NUCLEUS         1983Apr28 1981Dec16 PO  U     19069 19069   0   480  12   1 CYL        0
SYS1.DCMLIB          1983Apr28 1981Dec16 PO  U           19069   0   120   1   1 CYL        0
SYS1.INDMAC          1983Apr28 1981Dec16 PO  FB       80 19040   0   180   1   1 CYL        1
SYS1.CMDLIB          1983Apr28 1981Dec16 PO  U           19069   0   180  46   1 CYL        1
SYS1.HELP            1983Apr28 1981Dec16 PO  FB       80 19040   0   180  33   1 CYL        1
SYS1.SAMPLIB         1983Apr28 1981Dec16 PO  F        80    80   0   240 100   1 CYL        1
SYS1.MACLIB          1983Apr28 1981Dec16 PO  FB       80 19040   0  1500  47   1 CYL        1
SYS1.PROCLIB         1983Apr28 1981Dec16 PO  FB       80 19040   0   180   5   1 CYL        1
SYS1.TELCMLIB        1983Apr28 1981Dec16 PO  U           19069   0   120  31   1 CYL        1
SYS1.UADS            1983Apr28 --------- PO  FB       80   800   0    30  16   1 CYL        1
SYS1.VTAMLIB         1983Apr28 1981Dec16 PO  U           19069   0   120  11   1 CYL        1
SYS1.IMAGELIB        1983Apr28 1981Dec16 PO  U            1024   0    60   3   1 CYL        0
SYS1.PARMLIB         1983Apr28 1981Dec16 PO  F        80    80   0   180  36   1 CYL        0
SYS1.BRODCAST        1983Apr28 --------- DA  F            129   1    30 100   1 CYL        0
SYS1.MANX            1983Apr28 1981Dec16 PS  VBS    4096  4096   0   150  51   1 CYL        0
SYS1.MANY            1983Apr28 1981Dec16 PS  VBS    4096  4096   0   150   0   1 CYL        0
SYS1.TCOMMAC         1983Apr28 1981Dec16 PO  FB       80 19040   0  2970   0   1 CYL       10
SYS1.DUMP00          1983Apr28 1981Dec16     U             0   0   900   0   1 CYL        0
SYS1.LOGREC          1983Apr28 --------- PS  U        40    40   0    30  97   1 TRK        0
SYS1.VTAMSRC         1983Apr29 1981Dec16 PO  FB       80 19040   0    30   6   1 CYL        1
SYS1.VTAMOBJ         1983Apr29 --------- PO  F      3152  3152   0   150   1   1 CYL        5
SYS1.VTAMLST         1983Apr29 --------- PO  FB       80 19040   0   150   1   1 CYL        5
SYS1.UMODMAC         1983Apr29 1981Dec16 PO  FB       80 19040   0    30  68   1 CYL        1
SYS1.UMODSRC         1983Apr29 1981Dec16 PO  FB       80 19040   0    30  28   1 CYL        1
SYS1.UMODOBJ         1983Apr29 --------- PO  FB       80  3120   0    30   3   1 CYL        1
SYS1.UMODLIB         1983Apr29 1981Dec16 PO  U           19069   0    30   2   1 CYL        1
SYS1.SMPCRQ          1983Apr29 --------- PO  FB       80 19040   0    26  42   1 BLK      200
SYS1.SMPSCDS         1983Apr29 --------- PO  FB       80  3120   0    30   7   1 CYL        1
SYS1.COBLIB          1983Apr29 1981Dec16 PO  U           19069   0     6  87   1 TRK        0
SYS1.FORTLIB         1983Apr29 1981Dec16 PO  U           19069   0    14  88   1 TRK        0
SYS1.PL1LIB          1983Apr29 1981Dec16 PO  U           19069   0   122  99   1 TRK        0
SYS1.SORTLIB         1983Apr29 1981Dec16 PO  U           19069   0    21  83   1 TRK        0
```

**Figure 216: DASDLS utility output (extended information)**

## 8.15 DASDPDSU (PDS unload utility)

### *8.15.1 Function*

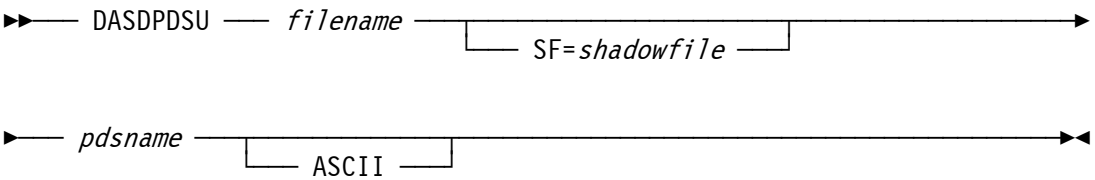
The DASDPDSU utility is a command that unloads PDS members from a DASD image and copies each member to a file *memname.*mac in the current working directory.

### *8.15.2 Syntax*

**Descriptive**

DASDPDSU *filename* [SF=*shadowfile*] *pdsname* [ASCII]

**Diagram**




### *8.15.3 Parameter*

**filename** Name of the DASD image file from which a PDS will be unloaded.

**shadowfile** Optional name of an associated shadow file.

**pdsname** Name of the partitioned dataset on the DASD image file from which the members will be unloaded.

**ASCII** If the optional keyword ASCII is specified the members will be unloaded as ASCII variable length text files. Otherwise the members are unloaded as fixed length EBCDIC binary files.

### *8.15.4 Examples*

**Example 1:**

Unload the dataset “SYS1.PARMLIB” on DASD image file MVSRES.CCKD in ASCII format.

```
DASDPDSU D:\MVS\DASD\MVSRES.CCKD SYS1.PARMLIB ASCII
```

### *8.15.5 Sample Utility Output*

**Sample 1:**

Unload the dataset “SYS1.PARMLIB” on DASD image file MVSRES.CCKD in ASCII format.

```
D:\Hercules>dasdpdsu D:\MVS\DASD\MVSRES.CCKD SYS1.PARMLIB ASCII
Hercules PDS unload program Version 3.05
(c)Copyright 1999-2006 by Roger Bowler, Jan Jaeger, and others
Reading directory block at cyl 198 head 0 rec 1
Member COMMND00 TTR=001A27
Member GTFPARM TTR=001A33
Member IEAABD00 TTR=001A35
Member IEAAPF00 TTR=001A29
Member IEABLD00 TTR=001B0C
Member IEADMP00 TTR=001A31
Reading directory block at cyl 198 head 0 rec 2
Member IEADMR00 TTR=001A2F
Member IEAIPS00 TTR=000112
Member IEALOD00 TTR=000712
Member IEAOPT00 TTR=000126
Member IEAPAK00 TTR=00062E
Member IEASYS00 TTR=002D2E
Member IPCSPR00 TTR=00041B
Member IRBMF100 TTR=001B1E
Member JES2PARM TTR=003C07
Reading directory block at cyl 198 head 0 rec 3
Member LNKLST00 TTR=001804
Member MVIKEY00 TTR=001B12
Member RPFKEY00 TTR=000B42
Member SMFPRM00 TTR=001B3B
Member TSOKEY00 TTR=001B14
Member VATLST00 TTR=001B45
End of directory
```

**Figure 217: DASDPDSU utility output**

## 8.16 DASDSEQ (Display sequential datasets)

### *8.16.1 Function*

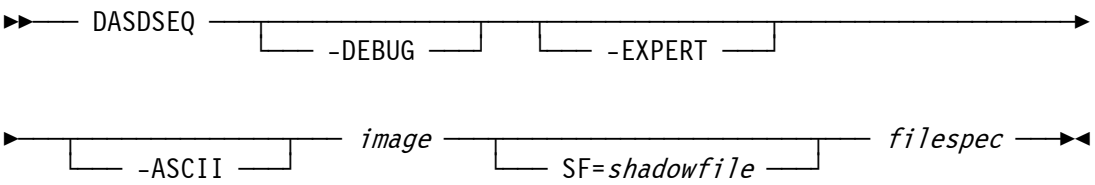
The DASDSEQ utility is used to extract and list sequential datasets from a DASD image file.

### *8.16.2 Syntax*

**Descriptive**

DASDSEQ [-DEBUG] [-EXPERT] [-ASCII] *image* [SF=*shadowfile*] *filespec*

**Diagram**




### *8.16.3 Parameter*

***Options:***

**-DEBUG** Display additional debug options.

**-EXPERT** Display an additional help panel that describes the expert operands.

**-ASCII** Translate the output file to ASCII and trim trailing blanks.

***Arguments:***

**image** The file name (and optionally path) of the DASD image file.

**shadowfile** The optional name (and path) of an associated shadow file.

**filespec** The name of a sequential dataset (DSORG=PS) on the DASD image file. This name is also used as filename for the extracted data.

All expert facilities (as described in the -EXPERT help panel) are experimental and therefore not yet described here.

## 8.16.4 Examples

**Example 1:**

Extract and list the sequential dataset SYS2.SEQ.DATA on the DASD image file TST001.AA0.

```
DASDSEQ -ASCII D:\MVS\DASD\TST001.AA0 D:\MVS\DASD\SYS2.SEQ.DATA
```

## 8.16.5 Sample Utility Output

**Sample 1:**

Extract and list the sequential dataset SYS2.SEQ.DATA on the DASD image file TST001.AA0.

```
D:\HERCULES>dasdseq -ascii D:\MVS\DASD\TST001.AA0 SYS2.SEQ.DATA
dasdseq 3.05 Copyright 1999-2005 Roger Bowler
Portions Copyright 2001-2005 James M. Morrison
dasdseq wrote 11 records to SYS2.SEQ.DATA
```

**Figure 218: DASDSEQ utility output**

## 8.17 HETGET (Extract files from an AWS or HET tape file)

### *8.17.1 Function*

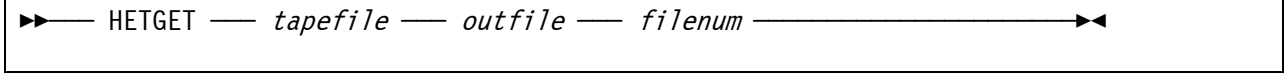
The HETGET utility extracts a file from an AWSTAPE or HET tape file. The output is in binary format and has to be displayed with an EBCDIC editor on the PC or has to be uploaded to the host operating system and opened with a mainframe editor.

### *8.17.2 Syntax*

**Descriptive**

```
HETGET tapefile outfile filenum
```

**Diagram**




### *8.17.3 Parameter*

**tapefile** The name (and optionally the path) of the AWSTAPE or HET tape file from which a file should be extracted.

**outfile** The name (and optionally the path) of the file to which the extracted file will be written.

**filenum** The file number on the tape of the file to be extracted from the tape.

### *8.17.4 Examples*

**Example 1:**

Extract file number 25 from the AWSTAPE file named T001003.AWS and write it to file FILE#25.BIN in directory D:/S390/UNLOAD/.

```
HETGET D:/MVS/TAPE/T001003.AWS D:/S390/UNLOAD/FILE#25.BIN 25
```

## *8.17.5 Sample Utility Output*

**Sample 1:**

Extract file number 10 from the AWSTAPE file named T38321A.AWS and write it to file FILE10.BIN.

```
D:\HERCULES>hetget D:/MVS/TAPE/T38321A.AWS D:/FILE10.BIN 10
Hercules HET extract files program Version 3.05
(c)Copyright 1999-2005 by Roger Bowler, Jan Jaeger, and others
File Info:
  DSN=BBC.BBC53XX.F5
DCB Attributes used:
  RECFM=V     LRECL=32756  BLKSIZE=32760
```

**Figure 219: HETGET utility output**

## 8.18 HETINIT (Initialize an AWS or HET tape file)

### *8.18.1 Function*

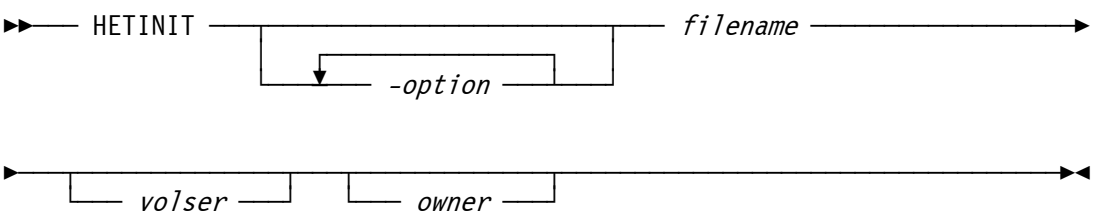
The HETINIT utility is used to initialize a tape file.

### *8.18.2 Syntax*

**Descriptive**

HETINIT [*-option* [*-option* ... ]] *filename* [*volser*] [*owner*]

**Diagram**




### *8.18.3 Parameter*

***Options:***

**-d** Disable compression (creates an AWSTAPE file).

**-h** Display usage summary (help text).

**-i** Create an IEHINITT formatted tape (set on by default!).

**-n** Create a NL (non labeled) tape.

***Arguments:***

**filename** Name (and optionally path) of the tape file to be created.

**volser** The volser of the tape.

**owner** The owner of the tape.

## 8.18.4 Examples

**Example 1:**

Create a new AWS tape file named NEWTAPE.AWS with volser NEWTAP and owner HERCULES in the directory D:/MVS/TAPE/.

```
HETINIT -d D:/MVS/TAPE/NEWTAPE.AWS NEWTAP HERCULES
```

## 8.18.5 Sample Utility Output

**Sample 1:**

Create a new HET tape file named NEWTAPE.HET with volser NEWTAP and owner HERCULES in the directory D:/MVS/TAPE/.

```
D:\HERCULES>hetinit D:/MVS/TAPE/NEWTAPE.HET NEWTAP HERCULES
Hercules HET IEHINITT program Version 3.05
(c)Copyright 1999-2005 by Roger Bowler, Jan Jaeger, and others
```

**Figure 220: HETINIT utility output**

## 8.19 HETMAP (Show information about a HET or AWS tape file)

### *8.19.1 Function*

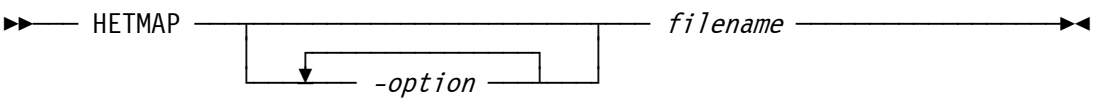
The Hercules HETMAP program displays information about a HET or an AWSTAPE tape file. By default, HETMAP shows label and file information. Optionally, HETMAP may be used to display data set information, or to display label information in a format similar to the output of TAPEMAP. HETMAP shows information about the tape and files on it, and does not display actual file contents.

### *8.19.2 Syntax*

**Descriptive**

HETMAP [*-option* [*-option* ... ]] *filename*

**Diagram**




### *8.19.3 Parameter*

***Options:***

**-a** Print all label and file information (default).

**-d** Print only dataset information.

**-f** Print only file information.

**-h** Display usage summary.

**-l** Print only label information.

**-t** Print TAPEMAP-compatible format output.

Options are mutually exclusive. If more than one option is entered, the last one is used. If no options are specified, "-a" is the default.

***Arguments:***

**filename** The name (and optionally path) of the HET or AWS tape file about which information is to be displayed.

## 8.19.4 Examples

**Example 1:**

Show information about a HET tape file.

```
HETMAP D:/MVS/TAPE/RPF142.HET
```

## 8.19.5 Sample Utility Output

**Sample 1:**

Show information about a HET tape file (print all label and file information).

```
D:\Hercules>hetmap -a G:\MVS\TAPE\RPF142.HET
Hercules HET and AWS tape map program Version 3.0.7
(c)Copyright 1999-2009 by Roger Bowler, Jan Jaeger, and others
--------------------
Filename            : G:\MVS\TAPE\RPF142.HET
--------------------
Label               : 'VOL1'
Volume Serial       : 'RPF142'
Improved Data Rec.  : ' '
Owner Code          : '          '
--------------------
Label               : 'HDR1'
Dataset ID          : 'RPF.INST.SRPFASM '
Volume Serial       : 'RPF142'
Volume Sequence     : '0001'
Dataset Sequence    : '0001'
GDG Number          : '    '
GDG Version         : '  '
Creation Date       : ' 74186'
Expiration Date     : ' 00000'
Dataset Security    : '0'
Block Count Low     : '000000'
System Code         : 'IBM OS/VS 370'
Block Count High    : '    '
--------------------
Label               : 'HDR2'
Record Format       : 'V'
Block Size          : '19060'
Record Length       : '19056'
Density             : '4'
Dataset Position    : '0'
Job/Step ID         : 'HERC01  /UNLOAD  '
Recording Technique : '  '
Control Character   : ' '
Block Attribute     : 'S'
Device Serial       : ' 30001'
Checkpoint ID       : ' '
Large Block Length  : '          '
--------------------
File #              : 1
```

```
Blocks              : 3
Min Blocksize       : 80
Max Blocksize       : 80
Uncompressed bytes  : 240
Min Blocksize-Comp  : 23
Max Blocksize-Comp  : 68
Compressed bytes    : 145
--------------------
File #              : 2
Blocks              : 196
Min Blocksize       : 20
Max Blocksize       : 19060
Uncompressed bytes  : 2907140
Min Blocksize-Comp  : 20
Max Blocksize-Comp  : 4739
Compressed bytes    : 579020
--------------------
Label               : 'EOF1'
Dataset ID          : 'RPF.INST.SRPFASM '
Volume Serial       : 'RPF142'
Volume Sequence     : '0001'
Dataset Sequence    : '0001'
GDG Number          : '    '
GDG Version         : '  '
Creation Date       : ' 74186'
Expiration Date     : ' 00000'
Dataset Security    : '0'
Block Count Low     : '000196'
System Code         : 'IBM OS/VS 370'
Block Count High    : '    '
--------------------
Label               : 'EOF2'
Record Format       : 'V'
Block Size          : '19060'
Record Length       : '19056'
Density             : '4'
Dataset Position    : '0'
Job/Step ID         : 'HERC01  /UNLOAD  '
Recording Technique : '  '
Control Character   : ' '
Block Attribute     : 'S'
Device Serial       : ' 30001'
Checkpoint ID       : ' '
Large Block Length  : '          '
--------------------
.
.
--------------------
File #              : 15
Blocks              : 2
Min Blocksize       : 80
Max Blocksize       : 8
Uncompressed bytes  : 160
```

```
Min Blocksize-Comp  : 54
Max Blocksize-Comp  : 73
Compressed bytes    : 127
--------------------
File #              : 16
Blocks              : 0
Min Blocksize       : 0
Max Blocksize       : 0
Uncompressed bytes  : 0
Min Blocksize-Comp  : 0
Max Blocksize-Comp  : 0
Compressed bytes    : 0
--------------------
Summary             :
Files               : 16
Blocks              : 425
Uncompressed bytes  : 3701156
Compressed bytes    : 913267
Reduction           : 2787889
```

**Figure 221: HETMAP utility output (print all label and file information)**

**Sample 2:**

Show information about a HET tape file (print only dataset information).

```
G:\Hercules>hetmap -d G:\MVS\TAPE\RPF142.HET
Hercules HET and AWS tape map program Version 3.0.7
(c)Copyright 1999-2009 by Roger Bowler, Jan Jaeger, and others
vol=RPF142             owner=

seq=1                  file#=2
dsn=RPF.INST.SRPFASM   crtdt=1974.186  expdt=0000.000  blocks=196
job=HERC01  /UNLOAD    recfm=VS        lrecl=19056     blksize=19060

seq=2                  file#=5
dsn=RPF.INST.SRPFOBJ   crtdt=1974.186  expdt=0000.000  blocks=126
job=HERC01  /UNLOAD    recfm=VS        lrecl=3136      blksize=3140

seq=3                  file#=8
dsn=RPF.INST.SRPFHELP  crtdt=1974.186  expdt=0000.000  blocks=8
job=HERC01  /UNLOAD    recfm=VS        lrecl=19056     blksize=19060

seq=4                  file#=11
dsn=RPF.INST.SRPFLOAD  crtdt=1974.186  expdt=0000.000  blocks=55
job=HERC01  /UNLOAD    recfm=VS        lrecl=19085     blksize=19089

seq=5                  file#=14
dsn=RPF.INST.CNTL      crtdt=1974.186  expdt=0000.000  blocks=19
job=HERC01  /UNLOAD    recfm=VS        lrecl=19056     blksize=19060
```

**Figure 222: HETMAP utility output (print only dataset information)**

**Sample 3:**

Show information about a HET tape file in TAPEMAP format.

```
D:\Hercules>hetmap -t D:\MVS\TAPE\RPF142.HET
Hercules HET and AWS tape map program Version 3.0.7
(c)Copyright 1999-2009 by Roger Bowler, Jan Jaeger, and others
VOL1RPF1420
HDR1RPF.INST.SRPFASM RPF14200010001       74186 000000000000IBM OS/VS 370
HDR2V190601905640HERC01  /UNLOAD      S   30001
File 1: Blocks=3, block size min=80, max=80
File 2: Blocks=196, block size min=20, max=19060
EOF1RPF.INST.SRPFASM RPF14200010001       74186 000000000196IBM OS/VS 370
EOF2V190601905640HERC01  /UNLOAD      S   30001
File 3: Blocks=2, block size min=80, max=80
HDR1RPF.INST.SRPFOBJ RPF14200010002       74186 000000000000IBM OS/VS 370
HDR2V031400313640HERC01  /UNLOAD      S   30001
File 4: Blocks=2, block size min=80, max=80
File 5: Blocks=126, block size min=60, max=3140
EOF1RPF.INST.SRPFOBJ RPF14200010002       74186 000000000126IBM OS/VS 370
EOF2V031400313640HERC01  /UNLOAD      S   30001
File 6: Blocks=2, block size min=80, max=80
HDR1RPF.INST.SRPFHELPRPF14200010003       74186 000000000000IBM OS/VS 370
HDR2V190601905640HERC01  /UNLOAD      S   30001
File 7: Blocks=2, block size min=80, max=80
File 8: Blocks=8, block size min=60, max=11232
EOF1RPF.INST.SRPFHELPRPF14200010003       74186 000000000008IBM OS/VS 370
EOF2V190601905640HERC01  /UNLOAD      S   30001
File 9: Blocks=2, block size min=80, max=80
HDR1RPF.INST.SRPFLOADRPF14200010004       74186 000000000000IBM OS/VS 370
HDR2V190891908540HERC01  /UNLOAD      S   30001
File 10: Blocks=2, block size min=80, max=80
File 11: Blocks=55, block size min=20, max=18864
EOF1RPF.INST.SRPFLOADRPF14200010004       74186 000000000055IBM OS/VS 370
EOF2V190891908540HERC01  /UNLOAD      S   30001
File 12: Blocks=2, block size min=80, max=80
HDR1RPF.INST.CNTL    RPF14200010005       74186 000000000000IBM OS/VS 370
HDR2V190601905640HERC01  /UNLOAD      S   30001
File 13: Blocks=2, block size min=80, max=80
File 14: Blocks=19, block size min=60, max=10672
EOF1RPF.INST.CNTL    RPF14200010005       74186 000000000019IBM OS/VS 370
EOF2V190601905640HERC01  /UNLOAD      S   30001
File 15: Blocks=2, block size min=80, max=80
File 16: Blocks=0, block size min=0, max=0
End of tape.
```

**Figure 223: HETMAP utility output (TAPEMAP compatible output)**

## 8.20 HETUPD (Update and/or copy an AWS or HET tape file)

### *8.20.1 Function*

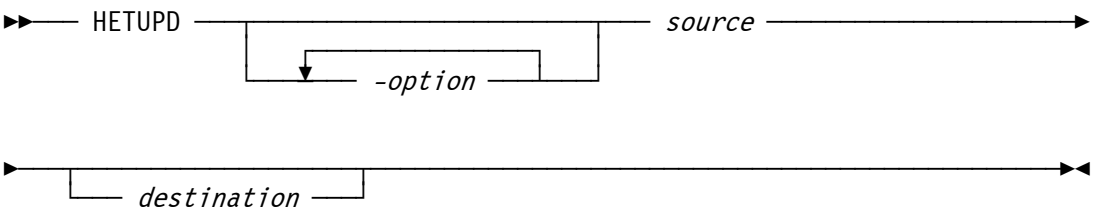
The HETUPD utility updates and/or copies emulated tape files.

### *8.20.2 Syntax*

**Descriptive**

HETUPD [*-option* [*-option* ... ]] *source* [*destination*]

**Diagram**




### *8.20.3 Parameter*

***Options:***

| | |
|---|---|
| **-1 … -9** | Compression level (1=compress fast, 9=compress best). |
| **-b** | Use bzlib compression. |
| **-c *n*** | Set chunk size to “n”. |
| **-d** | Decompress source tape file. |
| **-h** | Display usage summary (help text). |
| **-r** | Rechunk tape file. |
| **-s** | Strict AWSTAPE specification (chunksize=4096, no compression). |
| **-v** | Verbose information (display usage summary). |
| **-z** | Use zlib compression. |

***Arguments:***

| | |
|---|---|
| **source** | Name (and optionally path) of the source tape file. |
| **destination** | Name (and optionally path) of the destination tape file. |

## 8.20.4 Examples

**Example 1:**

Create an AWSTAPE file from an existing HET tape file.

```
HETUPD -s D:/MVS/TAPE/T001002.HET D:/MVS/TAPE/T001002.AWS
```

**Example 2:**

Change the compression level to 2 with BZLIB compression and create a new HET tape file.

```
HETUPD -2 -b D:/MVS/TAPE/TLEV009.HET D:/MVS/TAPE/TLEV002.HET
```

## 8.20.5 Sample Utility Output

**Sample 1:**

Change the compression level to 4 (best match between speed and compression rate) and create a new HET tape file.

```
D:\HERCULES>hetupd -4 D:/MVS/TAPE/T38321A.HET D:/MVS/TAPE/T38321B.HET
Hercules HET copy/update program Version 3.05
(c)Copyright 1999-2005 by Roger Bowler, Jan Jaeger, and others
```

**Figure 224: HETUPD utility output**

## 8.21 TAPECOPY (Copy a SCSI tape to or from an AWSTAPE disk file)

### *8.21.1 Function*

The Hercules tape copy program copies a SCSI tape to or from an AWSTAPE disk file. Tapecopy reads a SCSI tape and outputs an AWSTAPE file representation of the tape, or reads an AWSTAPE file and creates an identical copy of its contents on a tape mounted on a SCSI tape drive.

If the input file is a SCSI tape it is read and processed until physical EOD (end-of-data) is reached. That is, it does not stop whenever multiple tapemarks or filemarks are read, rather it continues processing until the SCSI tape drive says there is no more data on the tape.

The resulting AWSTAPE output disk file may be specified for the filename on a Hercules tape device configuration statement. It can then be used in order for the Hercules guest O/S to read the exact same data without having a SCSI tape drive physically attached to the host system. This allows you to easily transfer SCSI tape data to other systems that may not have SCSI tape drives attached to them.

The possible return codes and their meaning are:

0 Successful completion.

1 Invalid arguments or no arguments given.

3 Unable to open SCSI tape drive device file.

4 Unable to open AWSTAPE disk file.

5 Unrecoverable I/O error setting variable length block processing for SCSI tape device.

6 Unrecoverable I/O error rewinding SCSI tape device.

7 Unrecoverable I/O error obtaining status of SCSI device.

8 Unrecoverable I/O error reading block header from AWSTAPE disk file.

9 Unrecoverable I/O error reading data block.

10 AWSTAPE block size too large.

11 Unrecoverable I/O error writing tapemark.

12 Unrecoverable I/O error writing block header to AWSTAPE disk file.

13 Unrecoverable I/O error writing data block.

### *8.21.2 Syntax*

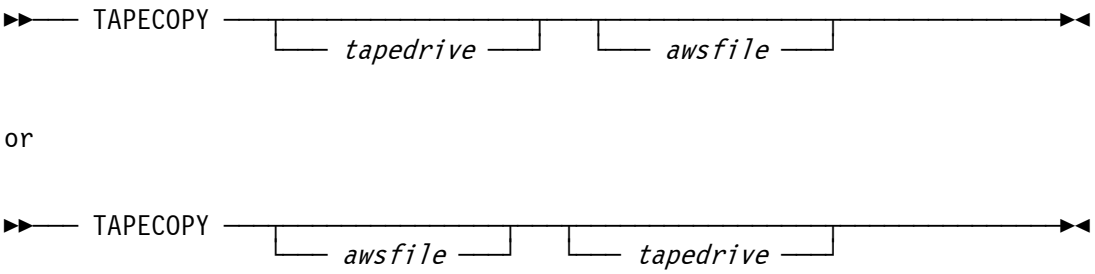
**Descriptive**

TAPECOPY [*tapedrive*] [*awsfile*]

or

TAPECOPY [*awsfile*] [*tapedrive*]

**Diagram**




## 8.21.3 Parameter

**tapedrive** Specifies the device filename of the SCSI tape drive. Must begin with */dev* to be recognized.

**awsfile** Specifies the filename of the AWSTAPE disk file.

The first filename is the input; the second one is the output.

## 8.21.4 Examples

**Example 1:**

Copy a SCSI file to an AWSTAPE disk file.

```
TAPECOPY /dev/nst0/scsifile.bin D:/MVS/TAPE/T3832AA.AWS
```

## 8.21.5 Sample Utility Output

**Sample 1:**

Copying an AWSTAPE disk file to a SCSI drive (this sample abends because the SCSI drive does not exist).

```
D:\HERCULES>tapecopy D:/T38321A.AWS /dev/nts0/scsifile.bin
Hercules tape copy program Version 3.05
(c)Copyright 1999-2005 by Roger Bowler, Jan Jaeger, and others
HHCTC001E Error opening /dev/nts0/scsifile.bin: errno=22: Invalid argument
HHCTC000I Abnormal termination
```

**Figure 225: TAPECOPY utility output**

## 8.22 TAPEMAP (Show information about an AWS tape file)

### *8.22.1 Function*

The TAPEMAP program shows information about an AWS tape. It displays all header data contained on the tape. Please note that the utility does not display the actual contents of the files. Although the TAPEMAP utility does display the same data as the HETMAP utility, the output is formatted in a totally different way.

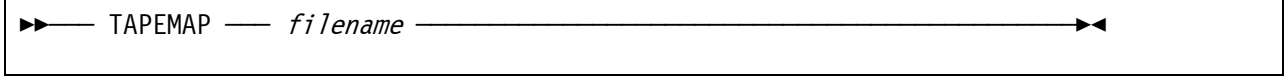
TAPEMAP produces correct output only for an AWS format tape. Use of TAPEMAP with any tape format other than AWS produces unpredictable results. Do not use it for HET tapes. The HETMAP program displays information about both AWS and HET tapes, and should be used instead of TAPEMAP.

### *8.22.2 Syntax*

**Descriptive**

TAPEMAP *filename*

**Diagram**




### *8.22.3 Parameter*

**filename** The name (and optionally path) of the AWS tape whose header data is to be displayed.

### *8.22.4 Examples*

**Example 1:**

Print map of an AWS tape file.

```
TAPEMAP D:/MVS/TAPE/T38321A.AWS
```

### *8.22.5 Sample Utility Output*

**Sample 1:**

Print map of an AWS tape file.

```
D:\Hercules>tapemap G:\MVS\TAPE\38321A.AWS
Hercules tape map program Version 3.0.7
(c)Copyright 1999-2009 by Roger Bowler, Jan Jaeger, and others
File 1: Blocks=614, block size min=800, max=800
File 2: Blocks=38, block size min=6272, max=6272
File 3: Blocks=90, block size min=160, max=800
File 4: Blocks=29, block size min=720, max=800
File 5: Blocks=113, block size min=80, max=800
File 6: Blocks=1581, block size min=800, max=800
File 7: Blocks=597, block size min=160, max=800
File 8: Blocks=181, block size min=320, max=800
File 9: Blocks=126, block size min=640, max=640
File 10: Blocks=11, block size min=560, max=800
File 11: Blocks=512, block size min=800, max=800
File 12: Blocks=0, block size min=0, max=0
End of tape.
```

**Figure 226: TAPEMAP utility output (map AWS tape file)**

## 8.23 TAPESPLT (Split an AWS tape file)

### *8.23.1 Function*

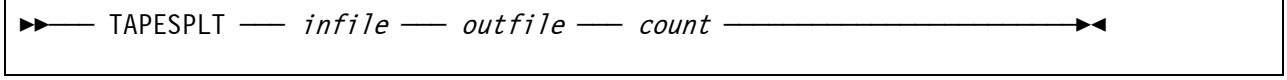
The TAPESPLT utility allows it to split an AWSTAPE file into a new AWSTAPE file. The number of the files to be copied defines the split point.

### *8.23.2 Syntax*

**Descriptive**

TAPESPLT *infile outfile count*

**Diagram**




### *8.23.3 Parameter*

**infile** The name of the input AWSTAPE disk file.

**outfile** The name of the output AWSTAPE disk file.

**count** The number of files that have to be copied to the output file.

### *8.23.4 Examples*

**Example 1:**

Copy the first three files from AWSTAPE file TOLDTAP.AWS to the new AWSTAPE file TNEWTAP.AWS.

```
TAPESPLT D:/MVS/TAPE/TOLDTAP.AWS D:/MVS/TAPE/TNEWTAP.AWS
```

### *8.23.5 Sample Utility Output*

**Sample 1:**

Copy the first 10 files from AWSTAPE file T38321A.AWS to T4352AA.AWS.

```
D:\HERCULES>tapesplt D:/MVS/TAPE/T38321A.AWS D:/MVS/TAPE/T4352AA.AWS 10
Hercules tape split program Version 3.05
(c)Copyright 1999-2005 by Roger Bowler, Jan Jaeger, and others
Writing output file D:/MVS/TAPE/T4352AA.AWS.
VOL138321A
```

```
HDR1DOCLIB           38321A00010001      0030140000000000000IBM OS/VS 370
HDR2V327523273600STP321VA/DOCLIB  P   S   68216
File 1: Blocks=3, block size min=80, max=80
File 2: Blocks=16, block size min=20, max=32740
EOF1DOCLIB           38321A00010001      0030140000000000016IBM OS/VS 370
EOF2V327523273600STP321VA/DOCLIB  P   S   68216
File 3: Blocks=2, block size min=80, max=80
HDR1RIMLIB           38321A00010002      0030140000000000000IBM OS/VS 370
HDR2V327523273600STP321VA/RIMLIB  P   S   68216
File 4: Blocks=2, block size min=80, max=80
File 5: Blocks=47, block size min=20, max=5140
EOF1RIMLIB           38321A00010002      0030140000000000047IBM OS/VS 370
EOF2V327523273600STP321VA/RIMLIB  P   S   68216
File 6: Blocks=2, block size min=80, max=80
HDR1HOLDDATA         38321A00010003      0030140000000000000IBM OS/VS 370
HDR2F031200008000STP321VA/HOLDATA P   B   68216
File 7: Blocks=2, block size min=80, max=80
File 8: Blocks=794, block size min=3040, max=3120
EOF1HOLDDATA         38321A00010003      0030140000000000794IBM OS/VS 370
EOF2F031200008000STP321VA/HOLDATA P   B   68216
File 9: Blocks=2, block size min=80, max=80
HDR1PGMDIR           38321A00010004      0030140000000000000IBM OS/VS 370
HDR2V327521065600STP321VA/PGMDIR  P   S   68216
File 10: Blocks=2, block size min=80, max=80
```

**Figure 227: TAPESPLT utility output**

## 8.24 DMAP2HRC (P/390 DEVMAP conversion program)

### *8.24.1 Function*

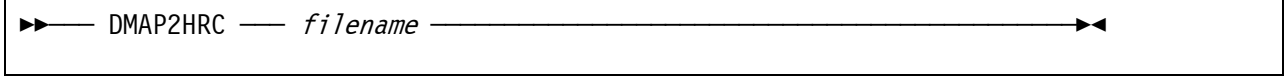
The P/390 DEVMAP to Hercules conversion program reads an IBM P/390 device map file, extracts the device definitions, and writes these as a valid Hercules configuration file to the standard output.

### *8.24.2 Syntax*

**Descriptive**

DMAP2HRC *filename*

**Diagram**




### *8.24.3 Parameter*

**filename** The name of the DEVMAP input file containing P/390 device statements.

### *8.24.4 Examples*

**Example 1:**

Convert the P/390 DEVMAP file DEVMAP.DVC to a Hercules configuration file named HERCULES.CNF.

```
D:/HERCULES/DMAP2HRC D:/S390/CONF/DEVMAP.DVC >D:/S390/CONF/HERCULES.CNF
```

### *8.24.5 Sample Utility Output*

With the exception of error messages (if any) the utility does not produce any screen output.

## 8.25 FTAPE (SCSI Tape Test Utility)

### *8.25.1 Function*

FTAPE is a command-line utility that allows to easily test the proper functioning of a Windows SCSI attached tape drive to make sure it's working properly before trying to add it to the Hercules device configuration. It supports functions like read and write, random or fixed-sized blocks of random data, fast-forward to the next or previous tape mark, back-space and forward-space blocks, write tape marks, locate blocks, etc. It will also display the data block it reads in either ASCII or EBCDIC to examine the data on an already existing tape.

It requires to already have the tape drive properly installed on the Windows system so that Windows can 'see' it (i.e. the SCSI adapter and SCSI tape device drivers are already installed such that device 'Tape0' appears in the Windows device list). When Windows can see the drive and FTAPE can access it, then the drive should be usable by Hercules.

Usually any “true” (non-ASPI) SCSI Tape device driver can be used, regardless of the tape drive model since all SCSI tape devices are required to support a minimum set of SCSI commands. The only exception found is IBM’s device drivers. They are purposely coded to only work with their own tape drives unfortunately.

Also note that while you need to use a non-ASPI driver in order for the drive to work with Hercules, that sometimes the ASPI software that comes with a tape drive also includes a device-driver for the tape drive itself, such that by installing whatever ASPI software may come with the tape drive, the needed non-ASPI device-driver also gets installed. In other words, some ASPI software packages include not only a control DLL that allows their software to talk to the tape drive via ASPI, but also includes the necessary device-driver for the tape drive itself.

Further note that FTAPE does not currently support doing any type of I/O to the medium changer device (i.e. the cartridge loader). It is no problem if the drive has one, it just cannot be directly accessed by FTAPE or Hercules.

### *8.25.2 Syntax*

**Descriptive**

FTAPE [-f *filename*] *function*

where *function* can be:

LOAD

UNLOAD

LOCK

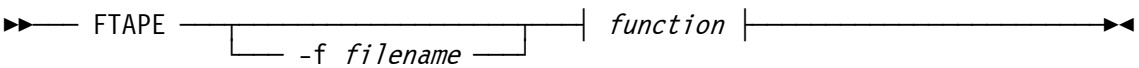
UNLOCK

REW

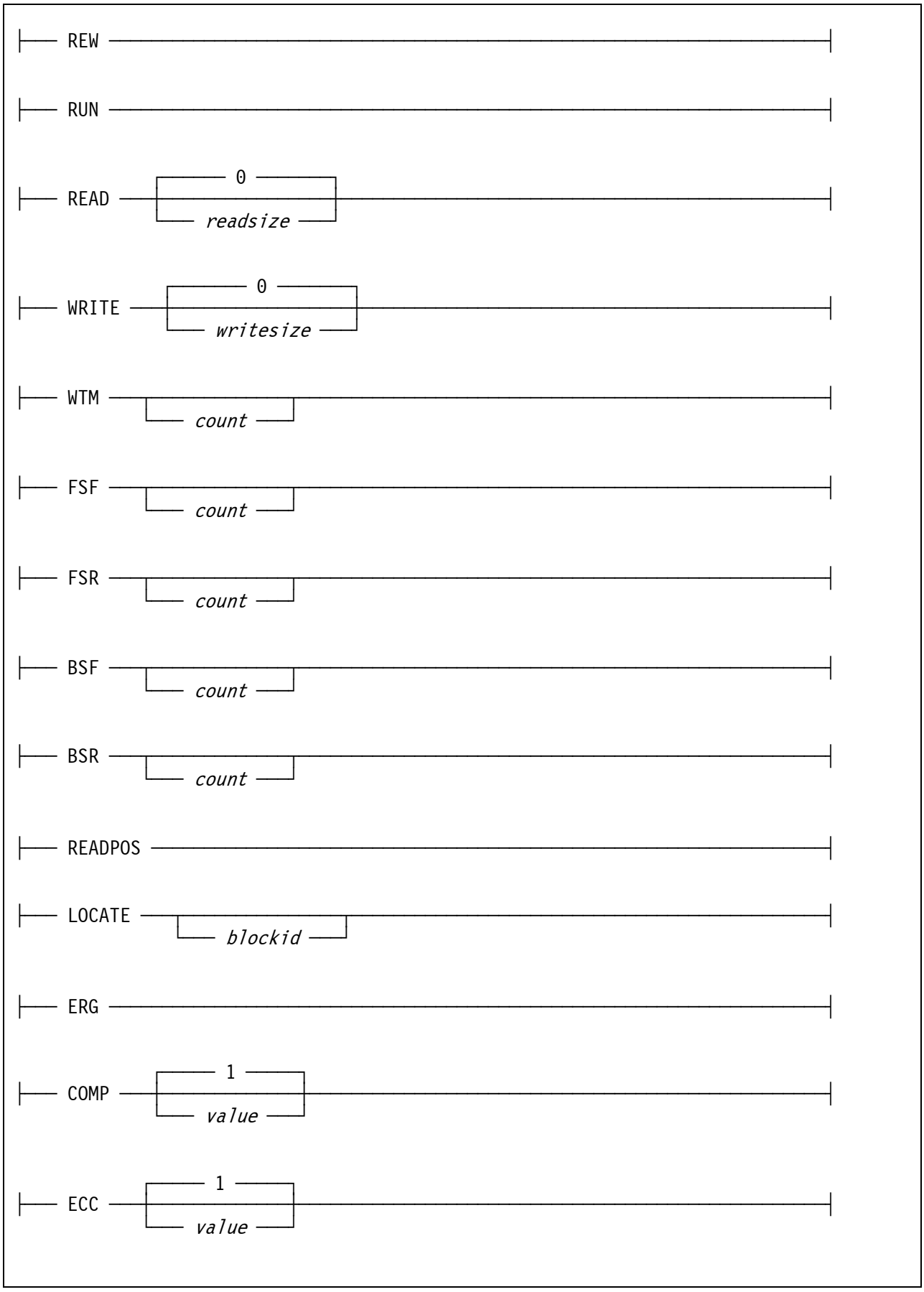
RUN

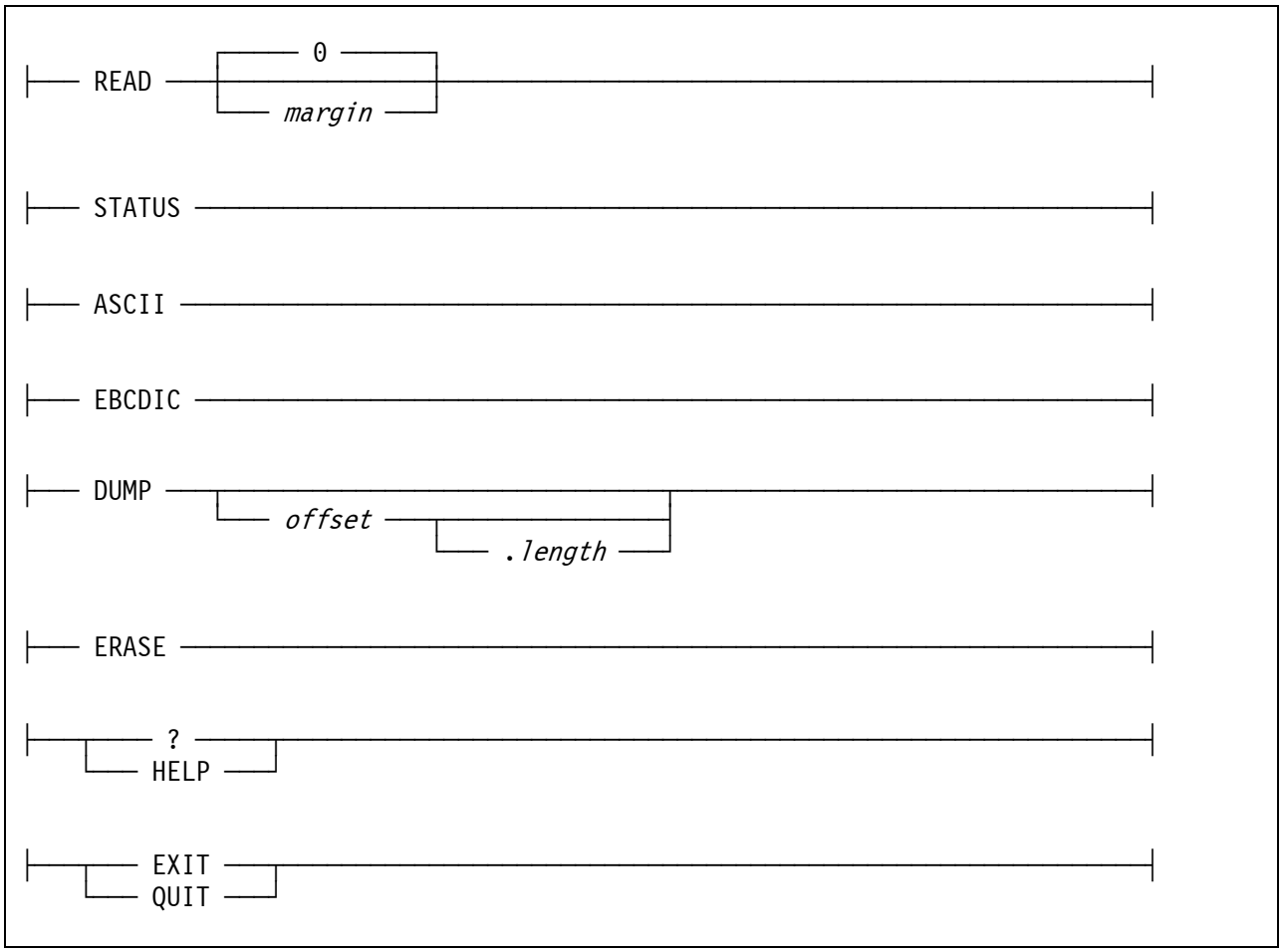
READ [*size* | 0]
WRITE [*size* | 0]
WTM [*count*]
FSF [*count*]
FSR [*count*]
BSF [*count*]
BSR [*count*]
READPOS
LOCATE [*blockid*]
ERG
COMP [*value* | 1]
ECC [*value* | 1]
MARGIN [*margin*]
STATUS
ASCII
EBCDIC
DUMP [*offset*[.*length*]]
ERASE
HELP | ?
EXIT | QUIT

**Diagram**




where *function* can be:


REW
RUN
READ
0
readsize
WRITE
0
writesize
WTM
count
FSF
count
FSR
count
BSF
count
BSR
count
READPOS
LOCATE
blockid
ERG
COMP
1
value
ECC
1
value




## *8.25.3 Parameter*

***filename*** This specifies the name of the tape device. The default is"Tape0".

***function*** The function that should be executed. For a list of available functions see below.

***functions:***

**LOAD** Load media.

**UNLOAD** Unload media.

**LOCK** Lock media.

**UNLOCK** Unlock media.

**REWIND** Rewind tape.

**RUN** Rewind and unload tape.

**READ** Read '*readsize*' blocks.

| | |
|---|---|
| **WRITE** | Write '*writesize*' blocks. |
| **WTM** | Write '*count*' tapemark(s). |
| **FSF** | Forward space '*count*' file(s). |
| **FSR** | Forward space'*count*' block(s). |
| **BSF** | Backward space '*count*' file(s). |
| **BSR** | Backward space '*count*' block(s). |
| **READPOS** | Read blockid. |
| **LOCATE** | Locate '*blockid*'. |
| **ERG** | Erase gap. |
| **COMP** | Set compression on or off. |
| **ECC** | Set ECC on or off. |
| **MARGIN** | Set EOT margin '*margin*'. |
| **STATUS** | Display status. |
| **ASCII** | Set ASCII data block display mode. |
| **EBCDIC** | Set EBCDIC data block display mode. |
| **DUMP** | Display data block at offset '*offset*' in length '*length*'. |
| **ERASE** | Erase entire tape (data security erase). |
| **HELP or ?** | Display usage information. |
| **EXIT or QUIT** | Exit program. |

***arguments:***

| | |
|---|---|
| ***readsize*** | Specifies the number of blocks to read. *readsize* must be a value from 0 to 65535. A value of 0 (which is the default) means all blocks, whereas any other number specifies the actual count of blocks to read. |
| ***writesize*** | Specifies the number of blocks to write. *writesize* must be a value from 0 to 65535. A value of 0 (which is the default) means random, whereas any other number specifies the actual count of blocks to write. |
| ***count*** | Specifies the number of files, blocks or tape marks to process. |
| ***blockid*** | Specifies the ID of the block to be located. |

***value*** Specifies the settings for compression and ECC. A value of '0' means OFF, whereas a value of '1' means ON. The default is '1' (ON).

***margin*** Specifies the EOT margin. *margin* must a value from 0 to 65535, the default is '0'.

***offset*** Specifies the offset from which data blocks are to be dumped.

***length*** Specifies the length of data blocks to be dumped.

## *8.25.4 Examples*

**Example 1:**

Display the status of SCSI tape device "Tape0".

```
FTAPE –f Tape0 STATUS
```

## *8.25.5 Sample Utility Output*

**Sample 1:**

Display the status of SCSI tape device "Tape0".

```
D:\Hercules>ftape

Fish's Win32 SCSI Tape Test Utility, v1.6.1.353
Copyright (C) 2004-2007, Software Development Laboratories
http://www.softdevlabs.com (fish@softdevlabs.com)

.
.
several lines not diplayed
.
.

Command? status

Retrieving status...

Warning! Unable to request variable blocksized i/o
         due to media parameters retrieval failure!



Tape device:  Tape0

Status:       Not Ready (Not Loaded)

Drive information:

ECC                          No
Compression                  No
```

```
DataPadding                  No
ReportSetmarks               No

DefaultBlockSize             32,768 bytes
MaximumBlockSize             65,536 bytes
MinimumBlockSize             1 bytes
MaximumPartitionCount        2 partitions
EOTWarningZoneSize           0 bytes

TAPE_DRIVE_ABS_BLK_IMMED     No
TAPE_DRIVE_ABSOLUTE_BLK      Yes
TAPE_DRIVE_END_OF_DATA       Yes
TAPE_DRIVE_FILEMARKS         Yes
TAPE_DRIVE_LOAD_UNLOAD       Yes
TAPE_DRIVE_LOAD_UNLD_IMMED   Yes
TAPE_DRIVE_LOCK_UNLOCK       Yes
TAPE_DRIVE_LOCK_UNLK_IMMED   No
TAPE_DRIVE_LOG_BLK_IMMED     No
TAPE_DRIVE_LOGICAL_BLK       Yes
TAPE_DRIVE_RELATIVE_BLKS     Yes
TAPE_DRIVE_REVERSE_POSITION  Yes
TAPE_DRIVE_REWIND_IMMEDIATE  Yes
TAPE_DRIVE_SEQUENTIAL_FMKS   No
TAPE_DRIVE_SEQUENTIAL_SMKS   No
TAPE_DRIVE_SET_BLOCK_SIZE    Yes
TAPE_DRIVE_SET_COMPRESSION   No
TAPE_DRIVE_SET_ECC           No
TAPE_DRIVE_SET_PADDING       No
TAPE_DRIVE_SET_REPORT_SMKS   Yes
TAPE_DRIVE_SETMARKS          Yes
TAPE_DRIVE_SPACE_IMMEDIATE   No
TAPE_DRIVE_TENSION           No
TAPE_DRIVE_TENSION_IMMED     No
TAPE_DRIVE_WRITE_FILEMARKS   Yes
TAPE_DRIVE_WRITE_LONG_FMKS   No
TAPE_DRIVE_WRITE_MARK_IMMED  Yes
TAPE_DRIVE_WRITE_SETMARKS    Yes
TAPE_DRIVE_WRITE_SHORT_FMKS  No
TAPE_DRIVE_COMPRESSION       No
TAPE_DRIVE_CLEAN_REQUESTS    No
TAPE_DRIVE_ECC               No
TAPE_DRIVE_EJECT_MEDIA       Yes
TAPE_DRIVE_ERASE_BOP_ONLY    No
TAPE_DRIVE_ERASE_LONG        No
TAPE_DRIVE_ERASE_IMMEDIATE   Yes
TAPE_DRIVE_ERASE_SHORT       Yes
TAPE_DRIVE_FIXED             No
TAPE_DRIVE_FIXED_BLOCK       Yes
TAPE_DRIVE_INITIATOR         Yes
TAPE_DRIVE_PADDING           No
TAPE_DRIVE_GET_ABSOLUTE_BLK  Yes
TAPE_DRIVE_GET_LOGICAL_BLK   Yes
TAPE_DRIVE_REPORT_SMKS       Yes
```

```
TAPE_DRIVE_SELECT             No
TAPE_DRIVE_SET_EOT_WZ_SIZE    No
TAPE_DRIVE_SET_CMP_BOP_ONLY   No
TAPE_DRIVE_TAPE_CAPACITY      No
TAPE_DRIVE_TAPE_REMAINING     No
TAPE_DRIVE_VARIABLE_BLOCK     Yes
TAPE_DRIVE_WRITE_PROTECT      Yes
.
.
.
```

**Figure 228: FTAPE utility output (display status)**

## 8.26 HercPrt (Remote Hercules Printer Spooler)

### *8.26.1 Function*

HercPrt is a remote Hercules printer spooler written by Fish (David B. Trout), the author of the widely used HercGUI, CTCI-WIN, AWSBrowse and FTAPE packages. HercPrt is designed to receive text output from a Hercules “socket device” (sockdev) line printer and use it to create a disk file on the local Windows system.

Because HercPrt uses standard TCP/IP sockets to communicate with a Hercules line printer, the actual Hercules system whose line printer output HercPrt is spooling, can be physically located at any place reachable via standard IP networking.

HercPrt supports creation of either plain text files, HTML files, Rich Text Format files (RTF) or Portable Document Files (PDF) according to the options provided in a “Job Separator Control File”. Several sample job separator control files are provided with the HercPrt package.

The job separator control file tells HercPrt what the Hercules guest operating systems job separator pages look like, so that it can automatically create separate Windows files for each print job, each optionally named according to a choice of job accounting field values extracted directly from the job separator page itself. HercPrt will create these spooled print files in whatever directory you choose using your specified options.

Several HercPrt instances can be run simultaneously, each spooling a different Hercules line printer for either the same instance or a completely different instance of Hercules. The options used for each printer are saved under a user specific printer ID so that the same options can easily be specified the next time HercPrt is run, by simply selecting the desired printer ID from a provided list of previously defined printer IDs.

The main dialog user interface is fully resizable and can be completely hidden (minimized) to the system tray area for minimal interference with normal Windows use. Automatic connection and reconnection support (with a user configurable delay between retries) is provided as well as complete control over the display of popup balloon tooltips used to notify the user of either incoming print output or loss of connectivity.

### *8.26.2 Installation*

The installation of HercPrt is beyond the scope of this manual. For detailed instructions on installing HercPrt on a Windows system please see the Hercules “Installation Guide”.

### *8.26.3 Hercules Configuration*

HercPrt is designed to receive text output from a Hercules “socket device” (sockdev) line printer. In order to use HercPrt with Hercules a “sockdev” printer must first be added to the Hercules configuration.

A Hercules socket printer definition looks similar to the following device definition statement:

```
000E 1403 192.168.0.100:14031 sockdev
```

The value ‘192.168.0.100:14031’ is the TCP/IP address and the port number at which the Hercules socket printer will listen for incoming connections. The IP address is typically the IP address of the Windows system where HercPrt is running on and the port number can be any value from 1024 to 65535.

Note that Hercules socket printers do not support any other options besides “sockdev”, the “clrf” and “noclear” options for example are invalid when defining a socket printer. For details on how to define printers in a Hercules configuration file please see the Hercules “User Reference” manual.

When Hercules is powered on the socket printer begins listening for incoming connections on the IP address and port number that are assigned to it. When you start HercPrt you define a printer at the same IP address and the same port number and click the connect button.

HercPrt should then establish a connection to that Hercules printer and immediately begins spooling reports from this printer if any are available.

Whether any reports are available or not on a given printer (as well as how to print them) depends on the guest operating system and associated spooler running under Hercules and is beyond the scope of this manual.

### *8.26.4 Program Options Page*

The following figure shows the first of two available configuration pages from the HercPrt application – the "Program Options" page.

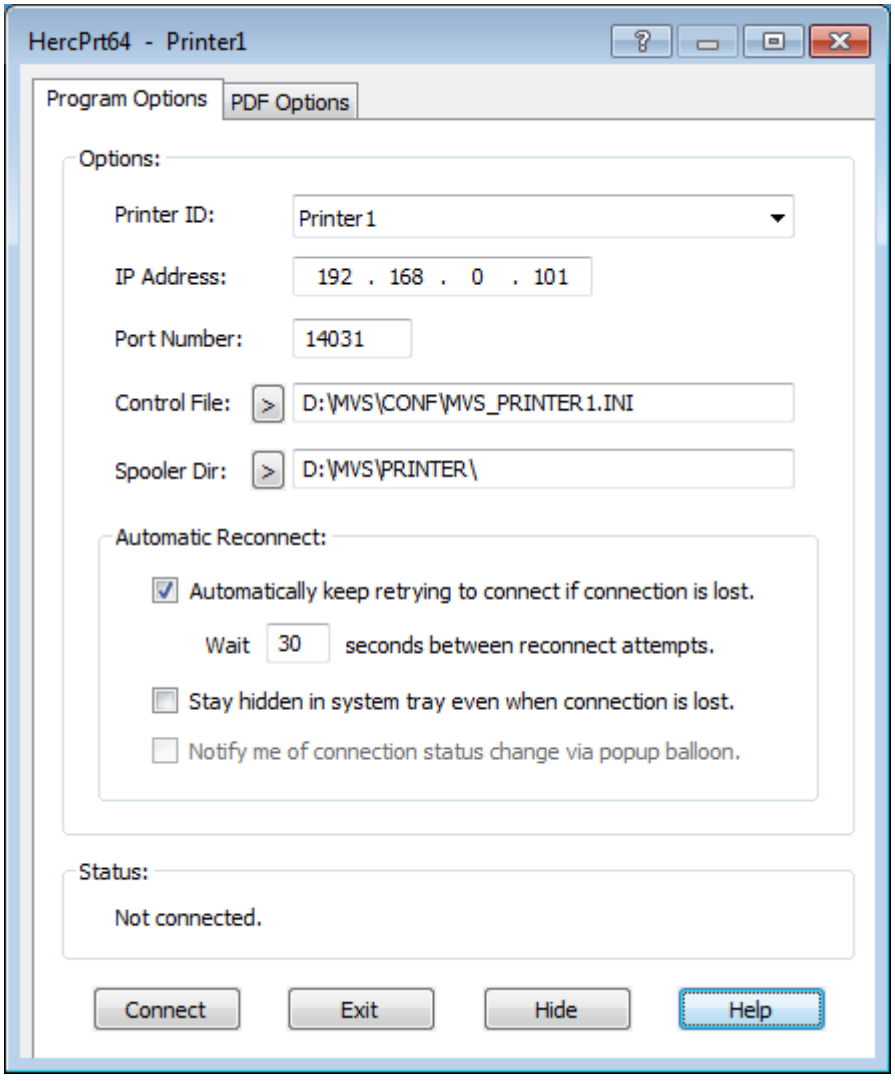



**Figure 229: HercPrt Program Options Page**

On the "Program Options" page you can configure the following:

**Printer ID** This field is used to enter a descriptive name for the Hercules printer you are defining. The value can be anything that uniquely identifies the printer being defined. If you already have some printers defined you can select one of these previously defined printers from the dropdown list to automatically populate the remaining controls with the values for the chosen printer.

**IP Address** This must be the IP address where your Hercules printer is listening for incoming connections.

**Port Number** This must be the port number where your Hercules printer is listening for incoming connections.

**Control File** This is the filename of the "Job Separator Control File" for the printer. The job separator control file defines what your Hercules guest operating system's job separator pages look like and allows HercPrt to detect where one print job ends and the next print job begins.

This allows HercPrt to break spooled output into separate Windows files, one for each printout, and to name the files using information from the job accounting fields extracted from the print fields on the actual job separator page itself.

Click on the arrow to display a standard Windows file open dialog where you can select the job separator control file you want to use for this printer.

**Spooler Dir** Enter here the name of the directory where you want the print files to be placed.

Click on the arrow to display a standard Windows file open dialog where you can select the directory for the print files of this printer.

**Automatic Reconnect** Check this box to automatically retry a failed connect attempt and to automatically keep retrying to reconnect whenever the connection is unexpectedly lost.

**Wait Seconds** This is the number of seconds to pause between the time a (re-)connect fails and the time the (re-)connect attempt is tried again. The idea here is not to bombard the Hercules system with a steady stream of (re-)connect attempts and to wait a short while before trying again to connect.

**Stay Hidden** Check this box to automatically hide the dialog once a connection has been made and to remain hidden even during reconnect retries whenever the connection is unexpectedly lost.

**Notify Me** Check this box to have the system tray icon automatically display a popup ballon whenever the connection status changes (e.g. connecting, receiving, idle, etc.).

**Status** This is the current status of the remote printer connection.

**Connect / Disconnect** Use this button to either connect to, stop retrying to connect to or disconnect from the specified Hercules printer.

The connect button will be labelled "Connect" if HercPrt is not already connected to a Hercules printer or else will be labelled "Disconnect" if HercPrt is connected to a printer or in the process to connect.

Clicking the button when it is labelled "Disconnect" will immediately disconnect from the specified printer if there is already a connection or else will immediately stop retrying to connect if HercPrt is currently in the process of connecting and "Automatic Reconnect" is enabled.

**Exit** Clicking this button immediately disconnects from the Hercules printer and exits the program.

**Hide** Clicking this button hides the dialog. To redisplay it again double click on the system try icon or right click and select “Show” from the appearing context menu. Single clicking the try icon display a popup balloon showing the current status.

**Help** Displays the programs help information.

## *8.26.5 PDF Options Page*

The next figure shows the second of two available configuration pages from the HercPrt application – the “PDF Options” page.

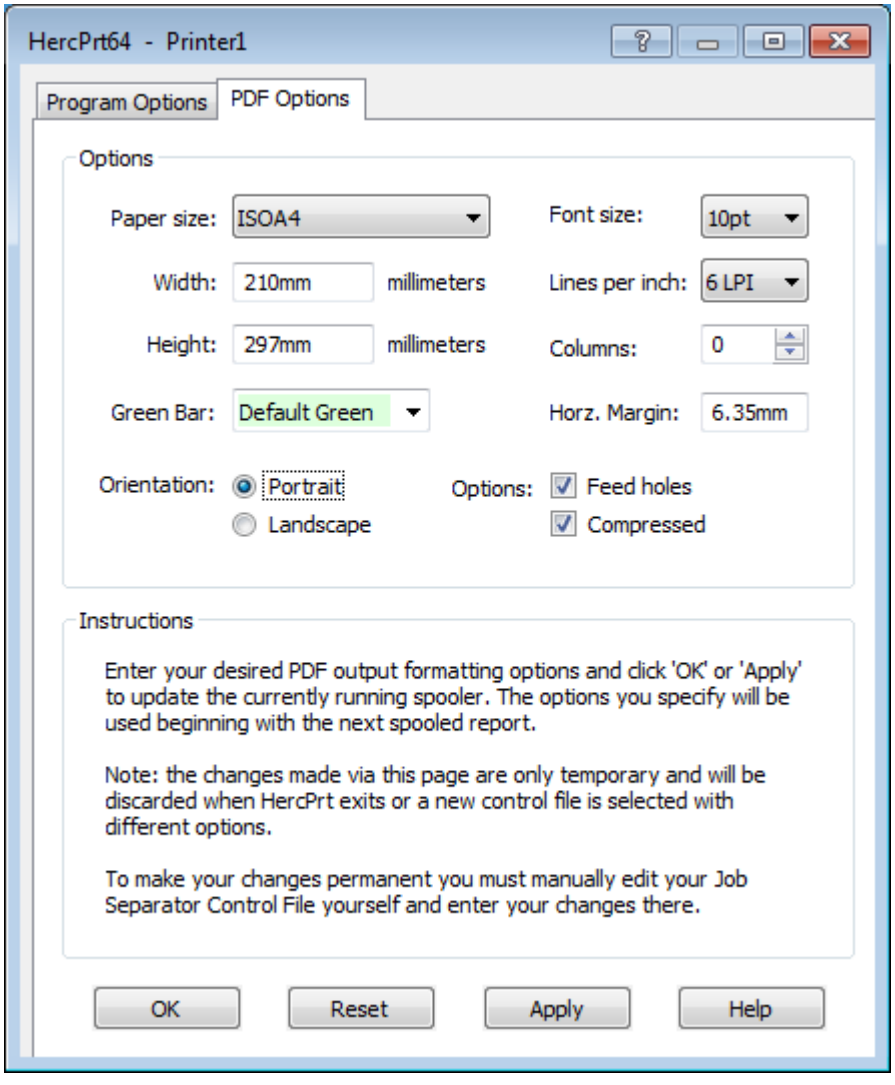



**Figure 230: HercPrt PDF Options Page**

On this page the desired PDF or RTF output formatting options can be specified to update the currently running spooler. The options specified here will be used beginning with the next spooled output.

The changes made on this page are only temporary and will be discarded when HercPrt exits or a new control file with different options is opened. To make this changes permanent you must manually edit the job separator control file and enter the changes there.

Please note that the options “Feedholes”, “Compression” and “Columns”, as well as all those related to the “Green Bar” are not supported for RTF but only to PDF files.

**Paper Size** This drop down allows to choose one of several predefined paper sizes. The pre-defined paper sizes are US Fan Fold, US Compact Fan Fold, US Letter, ISO A3, ISO A4, ISO B4, ISO B4 Extra and ISO C3.

**Paper Width** Enter here the desired custom form width in either fractional inches (the default) or whole millimetres. To specify millimetres append ‘mm’ to the size (e.g. ‘210mm’).

**Paper Height** Enter here the desired custom form height in either fractional inches (the default) or whole millimetres. To specify millimetres append ‘mm’ to the size (e.g. ‘297mm’).

**Green Bar** This dropdown is to select the desired “green bar” colour or a custom colour using the #RRGGBB format. Chose ‘No’ for no green bar or ‘Outline’ for an uncoloured “transparent” green bar.

Note: The green bar option is ignored for RTF output.

**Orientation** Choose the desired paper orientation, either portrait or landscape.

**Font Size** Use this dropdown to choose the desired font size from the list of available sizes. Available font sizes are 9pt, 10pt, 11pt and 12pt. Any size may be used for reports printed at 6 LPI, but the 9pt font size is forced for reports printed at 8 LPI.

The font itself is always “Courier New” and cannot be changed.

**Lines per Inch** The default line spacing is 6 lines per inch. If you choose to print your report at 8 LPI then the 9pt font size is forced.

**Columns** To squeeze long print lines onto narrow forms or stretch short print lines to fill wider forms enter here the number of columns you want to be able to fit onto your custom sized form.

The recommended default value of zero uses the normal default pitch of 10 characters (columns) per inch.

Note: The columns option is ignored for RTF output.

**Horizontal Margin** Specifies how far from the left edge of the form (not including the feed holes) you wish each print line to start.

When the output is PDF the same value is also used as the right margin and print lines are prevented from printing beond it.

When the output is RTF however, the value is only used as a left margin; the right margin is always set to zero and any print lines which would otherwise extend beyond the far right edge of the form are instead automatically wrapped onto the next line.

**Feed Holes** Select this option to cause a realistic looking set of tractor-feed holes to be drawn along the left and right sides of the report.

Note: The feed holes option is ignored for RTF output.

**Compressed** Produces standard compressed PDF files instead of legacy uncompressed ones,

thereby reducing the additional disk space the reports would otherwise consume.

Note: The compressed option is ignored for RTF output.

**OK** Clicking the “OK” button validates and saves the settings and then automatically hides the dialog. It is the same as clicking the “Apply” button followed by the (x) “Close” button.

**Reset** Clicking the “Reset” button causes the changes made on this panel to be discarded, resetting all values back to those defined in the [PDF] section of the job separator control file.

**Help** Displays the programs help information.

### *8.26.6 Help System*

Clicking the [?] context sensitive help button turns the mouse cursor into a question mark shaped cursor indicating the system is now in context sensitive help mode.

Once in context sensitive help mode you can the click with the mouse on any of the dialogs controls to cause a popup help balloon tooltip to be displayed containing a brief explanation of the control you clicked on.

To display the full help information for the control instead of just a popup balloon tooltip, then instead of clicking the [?] button first, simply press the F1 key while the control in question has the focus.

### *8.26.7 Tray Icon Context Menu*

The tray icon context menu appears whenever you right click on the HercPrt system tray icon. It has the following options:

**Show** Display the main dialog window. This is also the default action whenever the system try icon is double clicked.

**Connect** Selecting “Connect” performs the same action as the “Connect” button on the main dialog. It connects to the defined Hercules printer at the specified IP address and port number.

**Disconnect** Selecting “Disconnect” performs the same action as the “Disconnect” button on the main dialog. It either disconnects from the Hercules printer or stops trying to reconnect, depending on whether “Automatic Reconnect Retry” is enabled or not.

**Help** Opens the applications help file.

**About** Displays the applications “About” dialog, which displays the program version number and copyright information as well as the Software Developments Laboratories support email address used when submitting problem reports.

**Exit** Selecting “Exit” performs the same action as the “Exit” button on the main dialog. It immediately disconnects from the Hercules printer and exits the program.

## *8.26.8 Job Separator Control File*

The “Job Separator Control File” describes the layout of the guest’s job separator page as well as other spooler options. HercPrt uses this control file to be able to decide when the end of the spooled report has been reached as well as how to name the resulting file.

Additionally the control file also contains various options related to how you want your spooled output to look like for those file types whose appearance can be customized (currently limited to Portable Document Files (PDF) and to a limited degree to Rich Text Format (RTF) files).

HercPrt provides sample control files already configured for three of the most popular Hercules guest operating systems:

| **Control File Name** | **Operating System** |
|---|---|
| DOSVSR23.INI | DOS/VS Release 34 |
| VM370R6.INI | VM/370 Release 6 |
| MVSJES2.INI | MVS V3.8J |

**Table 20: Sample Control Files**

The provided samples can either be used as is or can be customized to suit your needs.

### 8.26.8.1 [Separator] section

The [Separator] section contains the information which defines what your guest's end-of-job or beginning-of-job separator page looks like. Each "trigger" defines one line of the separator page, and *all* of them must match together as a group in order for the page to be considered a job separator page.

For most systems these lines would be those which appear on your guest's end-of-job separator page, not the beginning-of-job separator page, since it is the *end*-of-job separator page which defines where the current print job ends and thus where the next one begins.

Some operating systems however use a spooler which only outputs beginning-of-job separator pages, making it impossible to know with any degree of certainty when the current print job has ended (except in the special case where another print job immediately follows it; the next print job's beginning-of-job separator would thus also serve to define the end of the previous job). Nevertheless a special format is supported to try and accommodate these unusual spoolers.

#### 8.26.8.1.1 Pages=

The first 'n' value of the "Pages=(n,m)" string specifies the number of consecutively identical end-of-job separator pages there are and defines the number of consecutive times in a row that all triggers must match.

The 'm' value defines the number of additional end-of-job separator pages there are following the identical pages. This parameter is needed to accommodate spoolers such as DOS/VS's POWER/VS spooler those separator pages are not all the same.

For MVS's JES2 spooler (whose separator pages are all the same) you should specify "(2,0)", whereas for DOS/VS's POWER/VS spooler (whose separator pages are not) the value "(2,1)" should be used instead. Both values of course presume the usual two page default. If your guest uses only one separator page then you should of course use (1,0) or (1,1) instead.

To accommodate non-standard spoolers such as VM/370 R6 which only output beginning-of-job separator pages but not any end-of-job separator pages (instead it just stops printing!), the "(n,TIMEOUT=x)" format is used instead.

The first number 'n' defines how many beginning-of-job separator pages to look for, and the 'x' in "TIMEOUT=x" defines how many seconds after which no more additional spooled output is received should indicate that the end of the print job has been reached.

Using a timeout value like this to indicate when the end of the print job has been reached is a somewhat unreliable way to detect the end of a print job, but without an end-of-job separator page it's the best that can be done for now.

#### 8.26.8.1.2 Triggers=

The "Triggers=" value defines how many trigger lines there are on your separator page. A trigger line is a line that exists on a job separator page that, together with other lines (triggers), identifies the page as a job separator page. The "Triggers=" value defines how many of these lines there are, and thus how many "[Trigger#]" sections there are following it.

#### 8.26.8.1.3 [Trigger#]

Each "[Trigger#]" section defines one separator page line. The "#" character in each section's name should of course be replaced with the associated trigger number. That is, if there are three lines on your job separator page then you would specify "Triggers=3" and define three "[Trigger#]" sections called "[Trigger1]", "[Trigger2]" and "[Trigger3]".

#### 8.26.8.1.4 Line=, Column=, Value=

The "Line=" value identifies which line of the separator page contains the text defined by the "Value=" value, and the "Column=" defines at which column of that line the text appears at. All line and column number values are 1-relative. Thus the first line on the page would be line 1 and the letter 'w' in "Hello world!" would be at column 7.

The text defined by "Value=" should be the exact text that appears at that line and column. All value comparisons are case sensitive and must match exactly. Enclose the value within double quotations if it contains any blanks.

There is no minimum or maximum value for the number of triggers you may define, but you should define enough of them to uniquely identify the page as a job separator page to prevent normal report pages from being erroneously detected as job separator pages.

### 8.26.8.2 [Field Names] section

The [FieldNames] section defines where the job accounting fields appear on your job separator page, thus allowing you to use them in your spooled report's output filename via the [OutputFile] section.

The individual job accounting field values are extracted directly from the separator page's individual print lines based on the values defined in the [FieldNames] section.

#### 8.26.8.2.1 Names=, Name#=

The "Names=" value defines how many job accounting fields are being defined and how many "Name#=" entries immediately follow it. The "#" character in each "Name#=" entry should of course be replaced with the associated field number. For example, if you wish to define three job accounting fields you would specify "Names=3" followed by "Name1=*one*", "Name2=*two*" and "Name3=*three*", where "*one*", "*two*" and "*three*" are of course the specific job accounting field's unique name.

You can assign whatever name you wish to your defined job accounting fields as long as the name does not contain any blanks and does not match any of the following reserved section names: "FieldNames", "OutputFile", "PDF", "Separator", "Translations" and "Trigger#".

#### 8.26.8.2.2 [fieldname] Line=, Column=, Width=

Each "[fieldname]" sub-section following the primary [FieldNames] section should match one of the field names defined in one of the "Name#=" entries. Note that it is not considered an error if it does not match any of your defined "Name#=" entries. Rather, it will simply never be used. The reverse however is not true. For each field name defined by a "Name#=" entry, a corresponding "[fieldname]" section must exist or it is considered an error.

The "Line=", "Column=" and "Width=" values define where on the page that particular field is located. Just like the "Line=" and "Column=" values of the [Separator] section, the line and column number values are 1-relative. A "Width=" value of 0 is invalid, but a width value of -1 means the field extends through to the end of the print line.

Whenever a job separator page is detected (as defined by the [Separator] section) all job accounting field values are extracted and saved according to the information contained in your [*fieldname*] sections. All defined job accounting field names are case sensitive and all extracted field values have their leading and trailing blanks removed before being saved.

### 8.26.8.3 [Output File] section

The [OutputFile] section contains a single "Name=" entry which defines how the job accounting fields must be used to construct the spooled report's filename. Job accounting fields are defined in the [FieldNames] section and their values are extracted directly from the job separator page and saved for use by the [OutputFile] section's "Name=" entry. Each percent sign enclosed string within the "Name=" value is substituted with one of the defined job accounting field values.

If, for example, have the following job accounting field defined in your [FieldNames] section and the contents of the job separator page's 58th line happens to be a "Name=" setting of "Job %JOBNAME%.txt" then this would result in the spooled report file being named "Job FOOBAR.txt".

Please note that Windows filenames cannot have any characters in them whose integer representation is within the range of 0 through 31 (i.e. x'00' to x'1F') nor can they contain any of the following reserved characters:

< > : " / \ | * ?

Further note that the length of a file's complete path specification cannot exceed the standard Windows operating system path length restriction of 260 characters total, so try to limit your output filename to the minimum needed while still ensuring that the resulting names remain different from one another.

After constructing an initial output filename based on your defined "Name=" value (which should be considered only a template), HercPrt then applies further string and character translations as defined in the [Translations] section before attempting to rename the file to its final name.

If during the construction of the output filename it is detected that a file already exists with that name, then "-2" is appended to the generated name and that is used instead. If a file with that name already exists then "-3" is appended, and so on, all the way to "-10". If a unique filename still cannot be constructed after trying "-10", then a warning is issued and the file is left with its original Windows assigned temporary filename (e.g. "PRT305.tmp").

The filename extension of your "Name=" value is used to determine what format the spooled file should be created in. If the "Name=" template ends with ".txt" the output file will be in plain text format (the default). If it ends with ".html" then it will be in HTML format, etc. Other supported filename extensions are ".rtf" and ".pdf". If the filename does not end with one of the supported filename extensions (txt, html, htm, rtf or pdf) then the file is created in text format by default.

If the output filename extension is either .rtf or .pdf then additional file formatting options may be specified via the [PDF] section.

### 8.26.8.4 [Translations] section

The [Translations] section allows you to further customize your output filename by allowing you to define character and string substitutions to take place during output filename construction.

#### 8.26.8.4.1 Strings=, Strings#-=, Strings#+=

Each "String#-=" value defines an output filename substring that is to be replaced with the string defined in the corresponding "String#+=" entry (where '#' represents a digit in the range of 1 to n with 'n' being the value defined by the preceding "Strings=" entry).

The "Strings=" value defines how many corresponding "String#-=" or "String#+=" entries there are. The replacement strings defined by the "String#+=" entries do not have to be the same length as the "String#-=" entry, and may even be empty to cause the substring defined in the "String#-=" entry to be completely removed.

#### 8.26.8.4.2 Chars-=, Chars+=

The "Chars+" value defines the replacement characters which should be used to replace the corresponding characters defined at the same relative position in the "Chars-" value. Each character of the filename generated by the "Names=" entry of the [OutputFile] section matching one of the characters defined in the "Chars-=" value is replaced with the corresponding character defined at the same relative position in the "Chars+=" value. There can only be one "Chars-= " and "Chars+=" entry but their string values (which must be the same length) can be as long as you wish.

In this way the "String..." and "Chars..." entries together allow you to define additional output filename transformations to take place during the construction of the final output filename. String replacement is always performed first followed by character replacement second, and a final internal character translation of:

```
< > : " / \ | * ?
```

to:

```
{ } . ' - _ ! @ !
```

is always performed last to ensure that no invalid or illegal characters remain in the final output Windows filename.

### 8.26.8.5 [PDF] section

The [PDF] section contains options used to define how you wish your output .pdf or .rtf file to be formatted. Please note that while all of the below options are supported for PDF output, not all of them are supported for RTF output.

The options which are not supported for .rtf files are: all of the GreenBar options (e.g. color) as well as the Feedholes, Compressed and Columns options. All of other PDF options (e.g. paper and font size, etc.) are supported for both PDF and RTF files:

| **Option** | **PDF** | **RTF** |
|---|---|---|
| PaperSize=*formname* | Yes | Yes |
| PaperWidth=*width* | Yes | Yes |
| PaperHeight=*height* | Yes | Yes |
| Orientation={Portrait \| Landscape} | Yes | Yes |
| HorzMargin=*margin* | Yes | Yes |
| FontSize={9 \| 10 \| 11 \| 12} | Yes | Yes |
| LPI={6 \| 8} | Yes | Yes |

| | | |
|---|---|---|
| Columns={0 \| *cols*} | Yes | No |
| GreenBar={No \| Yes} | Yes | No |
| BarColor={*colour* \| #*rrggbb*} | Yes | No |
| FeedHoles={No \| Yes} | Yes | No |
| Compress={Yes \| No} | Yes | No |

**Table 21: HercPrt PDF and RTF Options**

#### 8.26.8.5.1 PaperSize

Predefined "PaperSize=" values are "USFanFold" (default), "USCompactFanFold" (LPI=8), "USLetter", "ISOA3", "ISOA4", "ISOB4", "ISOB4Extra" and "ISOC3". When a predefined "PaperSize=" value is used then a "PaperWidth=" and "PaperHeight=" value need not be entered. Otherwise define your custom form size via "PaperWidth=" and "PaperHeight=".

#### 8.26.8.5.2 PaperWidth / PaperHeight

"PaperWidth=" and "PaperHeight=" values are expressed in either fractional inches (the default) or whole millimeters. To express a value in millimeters append "mm" to your value (e.g. "297mm").

Note: when defining your form's width do not include the width of the feedholes. The width of the feedholes (0.5 inches on either side of your form) is automatically added to your specified paper width at runtime whenever the "FeedHoles=Yes" option is specified. Your "PaperWidth=" value should always be specified as the actual printable width of your form and should not include the width of the feedholes area.

#### 8.26.8.5.3 Orientation

Specifying "Orientation=Landscape" or "Orientation=Portrait" will swap your specified "PaperWidth=" and "PaperHeight=" values if necessary in order to match your defined orientation.

Therefore defining "PaperWidth=8.5" and "PaperHeight=11" (or "PaperSize=USLetter") along with "Orientation=Landscape" will cause a paper width of 11 inches and a paper height of 8.5 inches to be used instead.

#### 8.26.8.5.4 HorzMargin

The "HorzMargin=" value specifies how far from the left edge of the form the first character of the print line is. For PDF output the "HorzMargin=" value defines both the left and right margins (the right margin being the end of the print line, i.e. the position past which no character will print). For RTF output however, it defines only the left margin and the right margin is always set to zero. Additionally, for RTF output, print lines that extend beyond the right margin automatically wrap to the next print line whereas for PDF output they are simply truncated.

The vertical margin at the top of every page is fixed at 0.5 inches and cannot be changed. The first printable line on a page after performing a page eject is therefore always the 4th physical line on the page when printing at 6 lines per inch or the 5th physical line if printing at 8 lines per inch (even though in both cases it is still considered to be line number 1).

The only way to print within the top 0.5 inches of your form is to continue printing beyond the bottom of the previous page such that your print then "overflows" onto the first physical line of the next page. Be aware however that you cannot do this continuously. That is to say your page cannot be of unlimited length. When any page grows to be 500 lines long or longer a page-eject is automatically forced and this behavior cannot be overridden.

#### 8.26.8.5.5 FontSize and LPI

"FontSize=" is always specified in typographic "points" (1/72 inches) with the default being 12pt for LPI=6 (Lines Per Inch) and 9pt for LPI=8. When 8 LPI is chosen then 9pt is forced. Valid font sizes are 9pt, 10pt, 11pt or 12pt.

The font that is used is always "Courier New" and cannot be changed.

#### 8.26.8.5.6 Columns

The "Columns=" value should normally be specified as 0 (zero) to ensure a proper default character pitch of 10 characters per inch. Specifying a nonzero "Columns=" value causes all characters on your print line to either be squished or stretched in an effort to try and honor your specified value, usually resulting in a very ugly looking printout.

#### 8.26.8.5.7 GreenBar, FeedHoles and BarColor

The default for "GreenBar=" and "FeedHoles=" is "No". "BarColor=" defaults to "Green". Other valid "BarColor=" values are: "Blue", "Yellow", "Red", "No" or "#rrggbb".

Specifying "BarColor=No" will draw an outlined box around the area where the green bars would normally be, but without being filled in with any background color (i.e. the color of the bar is set to "transparent"). Custom green bar colors can be specified using the "#RRGGBB" html RGB color value format.

#### 8.26.8.5.8 Compression

The “Compress=Yes” option (the default) performs normal ZLIB compression of the actual PDF page data stream in order to conserve Windows host disk space. “Compress=No” should only be used when reporting a problem viewing a HercPrt generated PDF file since such problems are usually the result of an improperly formatted PDF data stream and using ZLIB compression makes it difficult to see the actual PDF codes.

### *8.26.9 txt2pdf Utility*

HercPrt also ships with a command-line tool called "txt2pdf" which will convert a text file into either a PDF or RTF file.

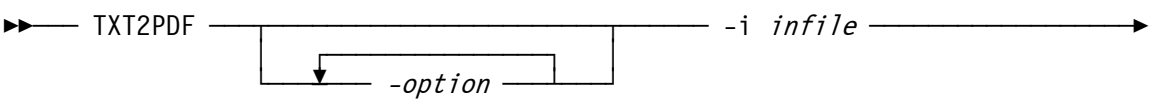
All of the PDF options that can be specified to HercPrt can also be specified to txt2pdf. Note however that just like with HercPrt, when converting text files to RTF files then not all of the PDF options are supported.

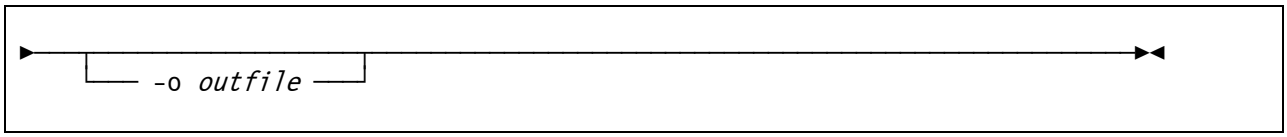
#### 8.26.9.1 Syntax

**Descriptive**

`TXT2PDF [-option [-option ... ]] -i infile [-o outfile]`

**Diagram**






### 8.26.9.2 Parameter

***options:***

**-a** Indicates that each input file line starts with an ASA/ANSI carriage control character (e.g. 1, 0, -, +, or <blank>).

**-l *lpi*** Lines per inch: 6 or 8. (The default is 6 LPI).

**-f *fontsize*** Font size: 9, 10 11 or 12 pt. (The default is 12 pt, if 6 LPI; otherwise 9pt).

**-w *width*** Paper width in inches or millimeters. (Default is 13.875 inches).

**-h *height*** Paper height in inches or millimeters. (Default is 11.0 inches).

**-m *margin*** Left / right margin in inches or millimeters. (Default is 0.25 inches).

**-t *"title"*** Document title property.

**-s *"subject"*** Document subject property.

**-r *"author"*** Document author property.

**-i *infile*** The name (and optionally path) of the input ".txt" file to be converted.

**-o *outfile*** The name (and optionally path) of the output ".pdf" or ".rtf" file (see notes).

***PDF options:***

**-g** Green bar paper.

**-c *color*** Green Bar color: 'Green' (default), 'Blue', 'Red', 'Yellow', or custom color specified as '#RRGGBB' HTML color value.

**-n** No color (i.e. outlined "transparent" green bar).

**-e** Feedholes.

**-u** Uncompressed PDF output. (Default is compressed).

**-# *cols*** Columns: 0-255. (Default is 0; see notes).

Notes:

- All PDF options are ignored when the output file is ".rtf" and not ".pdf".
- The -i switch is required in order to support specifying all options in any order. Options may precede the name of the input file or follow it or both, but the -i switch is required to identify the name of the actual input file.

- If -o is not specified the output file is named the same name as the input file but with a .PDF file extension. If you wish convert your file to .RTF instead of .PDF, then you MUST use the -o option to specify an output name ending with the ".RTF" file extension.
- The default paper size is US Standard Fanfold (13.875 x 11). Custom sizes should be specified in fractional inches or whole millimeters. To specify millimeters append 'mm' to your specified size (e.g. '287mm').
- The top margin size is fixed at 0.5 inches and cannot be changed.
- Do not include the width of the feedholes when specifying your paper width. The width of the feedholes (0.5 inches) is automatically added at runtime.
- The -n option is mutually exclusive with the -c option and overrides it when specified. Both are ignored if the -g option is not also specified.
- If the -# (columns) option is specified then each line is either squished or stretched in an attempt to fit exactly that number of columns of print onto each print line, usually resulting in a very ugly looking report. For best results it is recommended you not specify the -# option, or specify it as '0' (zero) for the default horizontal pitch of 10 characters per inch.

### 8.26.9.3 ASA Characters and their ASCII Equivalents

Unless the -a "ASA" option is specified, each line of the text input file that ends with a CR (carriage return) will cause that line to be overlaid with the text from the next one. Each line which ends with a LF (line feed or "new line" (NL) character) causes a skip to the next line after that line is printed. Skips to new pages only occur whenever a FF (form feed) character is encountered.

If the -a "ASA" option is specified, then each line of the input file is expected to start with an ASA (ANSI) carriage control character which identifies the action to be taken before that line is printed. Lines starting with a '1' cause a skip to the top of a new page before printing the line.

Lines starting with a blank cause a skip to a new line to occur before that line is printed. A '+' causes no spacing to occur thereby causing the line to overlay the previous one. '0' skips two lines before printing and '-' skips three lines before printing:

| ASA Character | Action to take before printing | ASCII Equivalent |
|---|---|---|
| 1 | Advance to next page (Form Feed). | FF |
| + | Do not advance any lines before printing (overstrike the previous line). | CR |
| (blank) | Advance one line (single spacing). | CR LF |
| 0 | Advance two lines (double spacing). | CR LF CR LF |
| - | Advance three lines (triple spacing). | CR LF CR LF CR LF |

**Table 22: HercPrt ASA Control Characters and ASCII Equivalents**

## *8.26.10 Samples*

The first sample shows a “Job Separator Control File” for MVS 3.8J. It is based on the sample that is provided with the HercPrt Software.

Please note, that the sections “Separator” and “FieldNames” have been shortened to keep the sample within reasonable limits in this manual. Both sections show only the first and the last entry of the nume-

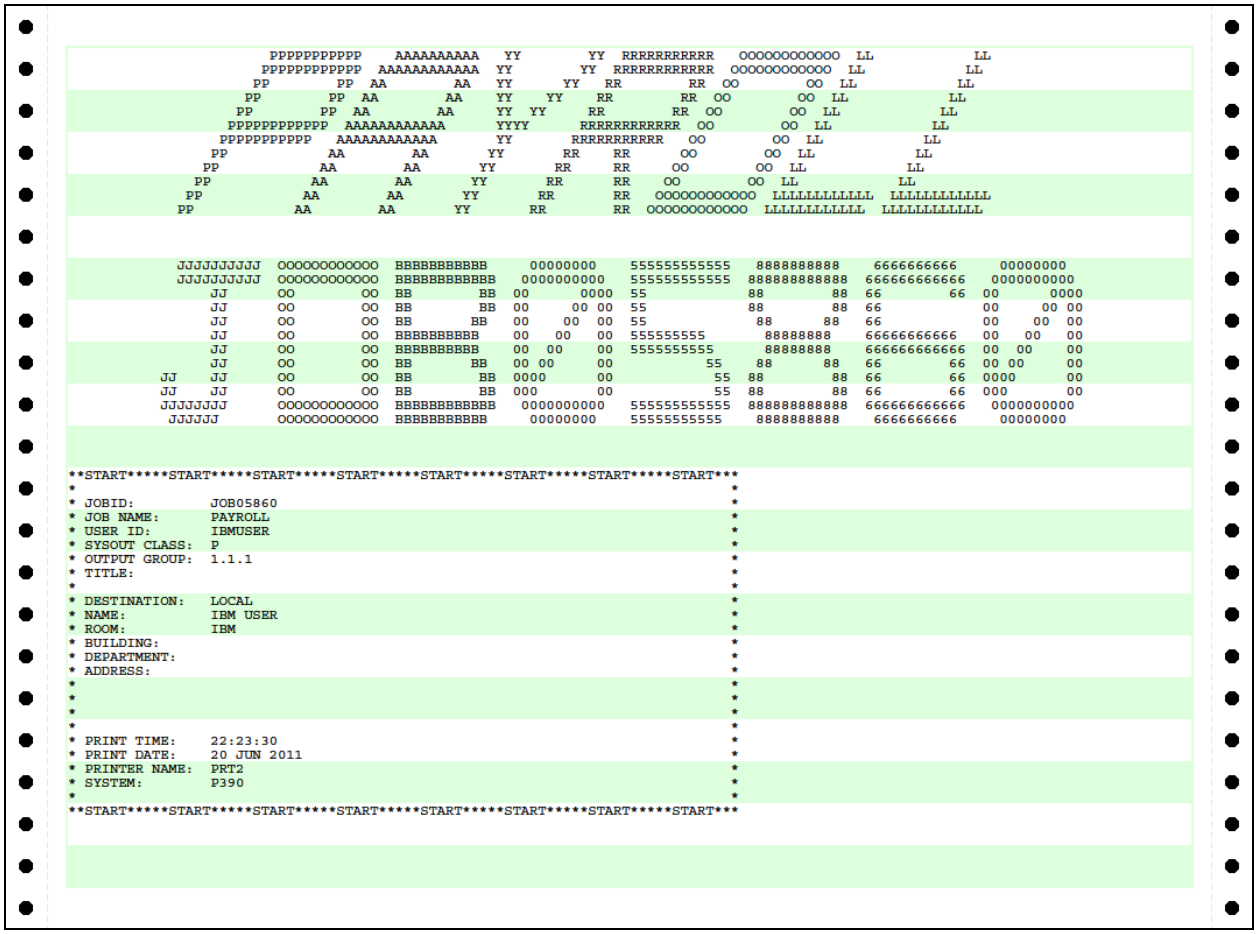
rous “Trigger” and “FieldNames” entries. Complete, working job separator control files are delivered with the HercPrt software for MVS 3.8J, for DOS/VS Release 34, and for VM/370 Release 6.

```
#-------------------------------------------------------------------------------
#                                Separator
#-------------------------------------------------------------------------------
[Separator]
Pages="(1,0)"
Triggers=25

[Trigger1]
Line=31
Column=1
Value="**END*******END*******END*******END*******END*******END*******END*******END*****"
.
.
.
[Trigger25]
Line=55
Column=1
Value="**END*******END*******END*******END*******END*******END*******END*******END*****"

#-------------------------------------------------------------------------------
#                              FieldNames
#-------------------------------------------------------------------------------
[FieldNames]
Names=16
Name1="JOBID"
.
.
.
Name16="SYSTEM"

[JOBID]
Line=33
Column=18
Width=62
.
.
.
[SYSTEM]
Line=53
Column=18
Width=62

#-------------------------------------------------------------------------------
#                              OutputFile
#-------------------------------------------------------------------------------
[OutputFile]
Name="%USERID%_%JOBNAME%_%JOBID%_%PRINTDATE%_%PRINTTIME%.pdf"
```

```
#-------------------------------------------------------------------------------------------
#                                Translations
#-------------------------------------------------------------------------------------------
[Translations]
Chars-="<>:"/\|?*"


Chars+="{}.'-_!!@"
Strings=6
String1-="      "
String1+=" "
String2-="     "
String2+=" "
String3-="    "
String3+=" "
String4-="   "
String4+=" "
String5-="  "
String5+=" "
String6-="PRT1"
String6+="000E"

#-------------------------------------------------------------------------------------------
#                            PDF / RTF Options
#-------------------------------------------------------------------------------------------
[PDF]
PaperSize=USFanFold
Orientation=Landscape
HorzMargin=0.25
FeedHoles=Yes
GreenBar=Yes
BarColor=Green
LPI=6
FontSize=12
Columns=0
Compress=Yes
```

**Figure 231: Job Seperator Control File Sample**

The second sample shows a page from a report created through HercPrt using the predefined job separator control file for MVS 3.8J.



```
                      PPPPPPPPPPP     AAAAAAAAAA   YY         YY  RRRRRRRRRRR   OOOOOOOOOOO  LL             LL
                     PPPPPPPPPPPP  AAAAAAAAAAAA  YY         YY  RRRRRRRRRRRR  OOOOOOOOOOO  LL             LL
                    PP         PP  AA        AA   YY       YY   RR         RR  OO         OO  LL             LL
                   PP         PP  AA          AA   YY     YY    RR         RR  OO         OO  LL             LL
                  PP         PP  AA            AA   YY   YY     RR         RR  OO         OO  LL             LL
                 PPPPPPPPPPPP  AAAAAAAAAAAA       YYYY       RRRRRRRRRRRR   OO         OO  LL             LL
                PPPPPPPPPPP   AAAAAAAAAAAA         YY        RRRRRRRRRRR    OO         OO  LL             LL
               PP           AA          AA        YY        RR     RR      OO         OO  LL             LL
              PP           AA            AA       YY        RR      RR     OO         OO  LL             LL
             PP           AA              AA      YY        RR       RR    OO         OO  LL             LL
            PP           AA                AA     YY        RR        RR   OOOOOOOOOOO  LLLLLLLLLLL  LLLLLLLLLLL
           PP           AA                  AA    YY        RR         RR  OOOOOOOOOOO  LLLLLLLLLLL  LLLLLLLLLLL


          JJJJJJJJJJ  OOOOOOOOOOO  BBBBBBBBBB     00000000    555555555555   8888888888    6666666666     00000000
          JJJJJJJJJJ  OOOOOOOOOOO  BBBBBBBBBBB   0000000000   555555555555  8888888888888  666666666666   0000000000
              JJ      OO         OO  BB        BB  00      0000  55              88        88  66        66  00      0000
              JJ      OO         OO  BB        BB  00     00 00  55              88        88  66            00     00 00
              JJ      OO         OO  BB       BB   00    00  00  55              88       88   66            00    00  00
              JJ      OO         OO  BBBBBBBBBB    00   00   00  555555555       88888888    66666666666   00   00   00
              JJ      OO         OO  BBBBBBBBBB    00  00    00  5555555555     88888888    666666666666  00  00    00
              JJ      OO         OO  BB        BB  00 00     00            55   88        88  66        66  00 00     00
        JJ    JJ      OO         OO  BB        BB  0000      00            55   88        88  66        66  0000      00
        JJ    JJ      OO         OO  BB        BB  000       00            55   88        88  66        66  000       00
        JJJJJJJJ      OOOOOOOOOOO  BBBBBBBBBBB   0000000000  555555555555  888888888888  666666666666  0000000000
         JJJJJJ       OOOOOOOOOOO  BBBBBBBBBB     00000000   55555555555    8888888888   6666666666    00000000


**START*****START*****START*****START*****START*****START*****START*****START***
*                                                                              *
* JOBID:          JOB05860                                                     *
* JOB NAME:       PAYROLL                                                      *
* USER ID:        IBMUSER                                                      *
* SYSOUT CLASS:   P                                                            *
* OUTPUT GROUP:   1.1.1                                                        *
* TITLE:                                                                       *
*                                                                              *
* DESTINATION:    LOCAL                                                        *
* NAME:           IBM USER                                                     *
* ROOM:           IBM                                                          *
* BUILDING:                                                                    *
* DEPARTMENT:                                                                  *
* ADDRESS:                                                                     *
*                                                                              *
*                                                                              *
*                                                                              *
*                                                                              *
* PRINT TIME:     22:23:30                                                     *
* PRINT DATE:     20 JUN 2011                                                  *
* PRINTER NAME:   PRT2                                                         *
* SYSTEM:         P390                                                         *
*                                                                              *
**START*****START*****START*****START*****START*****START*****START*****START***
```

**Figure 232: HercPrt Report Sample**

## 8.27 PRTPUB (Mainframe print publishing program)

The mainframe print publishing program is written by Dan Nelson and can be obtained in the file section of the Hercules-390 discussion group (http://groups.yahoo.com/group/hercules-390).

### *8.27.1 Function*

The mainframe print publishing program was designed to be used with Hercules' printing facilities. Hercules emulates line printers as simple text files containing printed output. This output can be viewed using a text editor or some other mechanism (such as being served via a web server).

This PRTPUB program converts the raw printer output to a more readable form by replacing formfeed characters with line break lines, converting newlines to CR/LF pairs if desired (for Windows viewing), and placing the result in a specified file.

Several facilities are provided to allow for publishing the printed output in an automated way (e.g. to a web server) as follows.

1) You can choose to publish the output only if the input file size has changed since the last time the program was run. The file size is stored in a file for comparison on each run and the program terminates with return code 4 if the input file size has not changed since the last run. This allows efficient mirroring of the printed output to a separate file on a schedule, only updating the target file if the source file has changed. There is also an option that permits the polling for change to be carried out in this program, thus saving the resources that would otherwise be used to launch the program, parse the options, and so forth on each iteration.

2) You can choose to publish only the last part of the printed output. Hercules printer files tend to grow and grow in size and users are typcially only interested in the last part, printed most recently. An option lets you specify the number of lines at the end of the file to publish to the target file, or lets you publish only the new data since the last run.

3) Locking is provided under UNIX type operating systems. As a courtesy this program will acquire an advisory read lock on the entire file before it starts processing. If the emulator uses an advisory write lock then this program can be scheduled to publish the output asynchronously to the emulator producing that output and the result will always be consistent.

4) If you want to publish the output as a web page without using a mechanism like server side includes, this program provides the option of supplying HTML tags to wrap the output, although this facility is only rudimentary. For sophisticated formatting server side includes are likely more appropriate to pull data into a page but this facility may be useful for simple scenarios.

The program should be used in a batch file. It is possible to use it for a single call or to build a loop for continuous processing (see examples below).

The possible return codes and their meaning are:

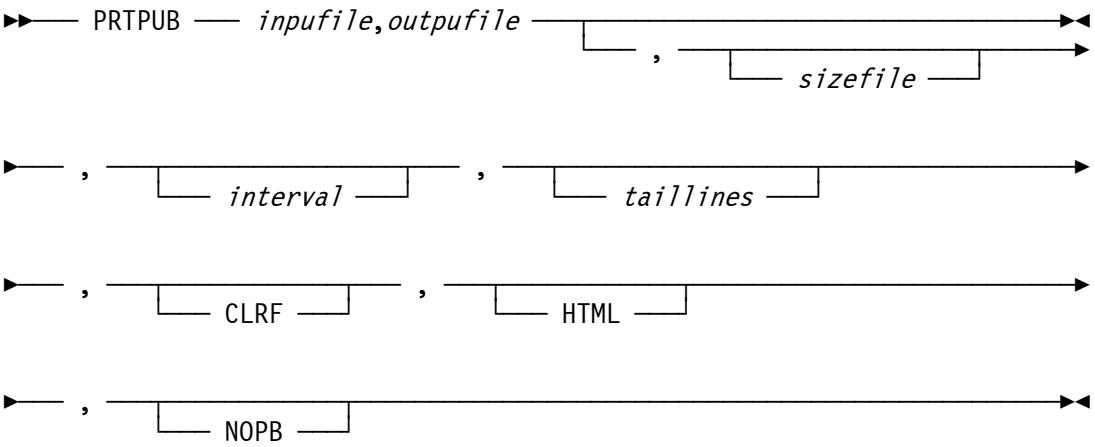
| | |
|---|---|
| 0 | Translation was done. |
| 4 | Translation was skipped because the file size did not change. |
| 8 | Processing failed for an ordinary reason. An error message is sent to stderr indicating why. |
| 16 | Processing failed due to some abnormality. An abend code is given for diagnostic purposes. This code will be the source line where the error occurred so you can use it with the source listing to diagnose the cause of the failure. |

## 8.27.2 Syntax

**Descriptive**

```
PRTPUB inputfile,outputfile [,[sizefile] ,[interval] ,[taillines]
                            ,[CRLF] ,[HTML] ,[NOPB]]
```

**Diagram**




## 8.27.3 Parameter

The program takes a single long argument which must not contain spaces. If the file paths (on Windows) contain spaces then the entire argument must be enclosed in quotes. No spaces are allowed anywhere else, whether or not quotes are used. If quotes are used then they are only permitted as the first and last characters of the argument string.

The argument is divided into subparameters separated by commas. All parameters are positional, if they are omitted a comma must mark their place.

**inputfile** The file containing the emulated mainframe output.

**outputfile** The file to which the result is to be published.

**sizefile** The file containing the input file's size at the time of the previous run. If specified, this file is used to track the size of the input file, only publishing when it changes. The file will generally not exist the first time it is specified. In that case, the publishing is done and the file is written for use in subsequent runs. The file contains a number in binary form. It cannot be directly edited. If this parameter is not specified then the publishing is always done regardless of the file's size.

**interval** The polling interval for use with the size file. If this argument is specified it must be a positive integer and it indicates a number of seconds between file size checks. In this mode if the input file's size has not changed since the last invocation (based on the size stored in the size file), the program goes into polling mode where it continually checks the size every interval seconds until a change is detected and then continues processing. If

this argument is not given the default behavior is to return with a return code of 4 when the input file's size has not changed. Using this argument can improve the efficiency of simple polling in cases where nothing else is to be done on each polling interval. This is because it does not require the program to be invoked, arguments to be processed etc., on each polling interval. If no size file is specified this argument is checked for syntax but otherwise ignored.

**taillines** The number of lines that should be published. If the input file contains more than the given number of lines then only the last <tail lines> lines are published. If this parameter is not specified the entire file is published. This number gives the number of printed lines published, including page separators. Note that the number may be off by one in either direction. If the final character of the file is a formfeed then a trailing page separator is always output, which can add an extra line in some cases. By the same token if the tail is broken on a formfeed character then the entire formfeed, which counts as two line breaks, is skipped, resulting in one fewer line of output. If this argument is 'ONLY' (in all caps) then the size file must also be specified. In that case the program translates only the part of the printer file that is new since the prior invocation, unless the new size is smaller than the old size, in which case the whole file is translated. The purpose of this feature is to allow this program to be used as a sort of spool writer in conjunction with scripts that do something interesting with the incremental output, such as print it to a printer. In this mode empty output files are not produced. If the output is zero length and ONLY is specified, the program terminates with a return code of 4.

**CRLF** If specified then all linefeeds in the source file are translated to CR/LF pairs and CR/LF is used as the terminator for the page break line. If not specified no linefeed translation is done and the page break line is terminated with a newline character. This parameter is case sensitive (must be uppercase).

**HTML** If specified then HTML tags are added to the published output file. The file begins with <HTML><PRE> and ends with </PRE></HTML>. This parameter is case sensitive (must be uppercase).

**NOPB** If specified then formfeed characters are not translated to page break lines. This is useful primarily for processing the published data after it is published, e.g. sending it to a real printer, etc. This argument must be in upper case. When NOPB is specified formfeed characters are still bracketed with newlines to keep the maximum output record size consistent.

## *8.27.4 Examples*

**Example 1:**

Publish the print output from print file PRT1.TXT and create a Word document named PRT1.DOC with carriage return/line feeds and do not translate formfeed characters to page break lines.

```
D:\HERCUELS\PRTPUB D:\MVS\PRT\PRT1.TXT,D:\MVS\PRT\PRT1.DOC,,,,CRLF,,NOPB
```

**Example 2:**

Publish the print output from print file PRT2.TXT and create a HTML document named PRT2.HTML with carriage return/line feeds.

```
D:\HERCULES\PRTPUB D:\MVS\PRT\PRT2.TXT,D:\MVS\PRT\PRT2.HTML,,,,CRLF,HTML
```

**Example 3:**

Publish the print output from print file PRT00E.TXT and create a HTML file called PUBL.HTML with carriage retun/line feeds. The input file's size at the time of the previous run should be kept in file PRT00E.SIZ. The polling interval to check the file size is 10 seconds. The maximum number of lines that should be published is 1000.

```
@echo off
:LOOP
prtpub prt00e.txt,publ.html,prt00e.siz,10,1000,CRLF,HTML
if errorlevel 1 goto NOTDONE
echo PRINT FILE UPDATED
goto LOOP
:NOTDONE
echo ERROR - NO UPDATE DONE
goto LOOP
```

## *8.27.5 Sample Utility Output*

With the exception of error messages (if any) the utility does not produce any screen output.

## 8.28 AWSSL (AWS virtual tape)

AWSSL is written by Reed H. Petty and is available on the CBT MVS Utilities Tape (CBTTAPE) at www.cbttape.org (File #585).

### *8.28.1 Function*

AWSSL moves MVS (OS390, etc) datasets directly to/from AWS (and HET) virtual tape files. AWSSL strives to maintain compatibility with other AWS/HET utilities including AWSUTIL, VTTS, HETUTL and the Hercules Emulator.

AWSSL creates AWS and HET format virtual tapes which contain one to many datasets of any record format, with or without standard labels. The program will also retrieve datasets from AWS/HET files of any record format with the exception of spanned blocks. The retrieved datasets may be reblocked if necessary. If DCB attributes are omitted on the receiving dataset and if standard labels are present within the AWS structure, then the DCB attributes of the receiving dataset will be defaulted to those within the HDR1 label.

### *8.28.2 Input Parameter*

All input parameters are taken from control statements supplied by the AWSCNTL DD statement. The statements consist of a major function to be performed (i.e. TAPEVOL, EXPORT, etc.), and keywords which supply values to that function.

Control statement keywords may be continued to as many records as necessary. Continued statements are indicated by the last keyword argument suffixed with a comma and additional keywords on the next record. Additional keywords must not begin in column one.

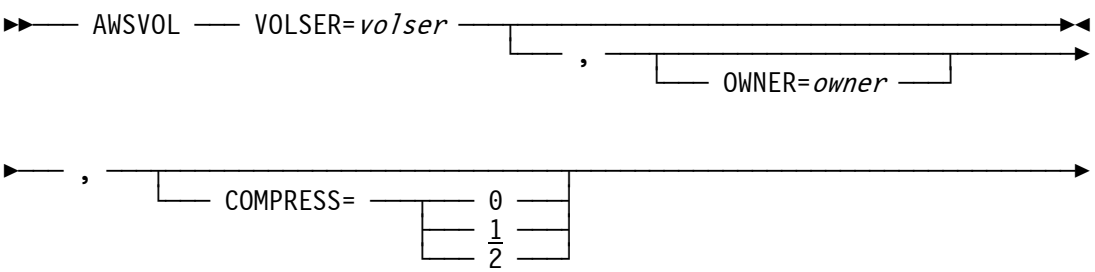
### *8.28.3 AWSVOL Control Statement*

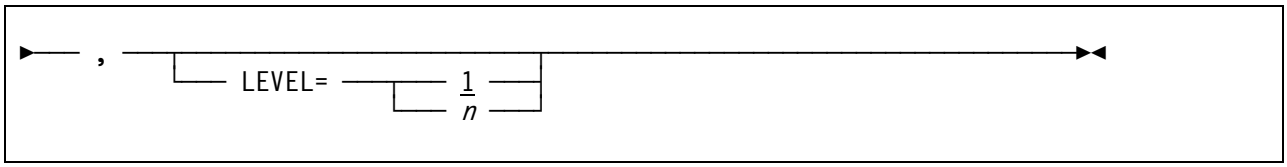
The AWSVOL control statement supplies characteristics of the virtual tape volume including volume serial number, owner and compression. AWSVOL must be the first control statement specified.

**Descriptive**

```
AWSVOL VOLSER=volser [,OWNER=owner] [,COMPRESS={0 | 1 | 2}] [,LEVEL={1 | n}]
```

**Diagram**






**VOLSER=volser** This specifies the volume serial number, volser is a 1 to 6 byte argument.

**OWNER=owner** The given *owner* will be included into the VOL1 owner field, owner is a 1 to 10 byte argument.

**COMPRESS=n** Compression, 0 = no compression, 1 = zlib, 2 = bzip. If COMPRESS=0 is coded then an AWS format is assumed. Default is HET format with COMPRESS=1 (zlib).

**LEVEL=n** Specifies the level of compression (0 – 9). Default is LEVEL=1.

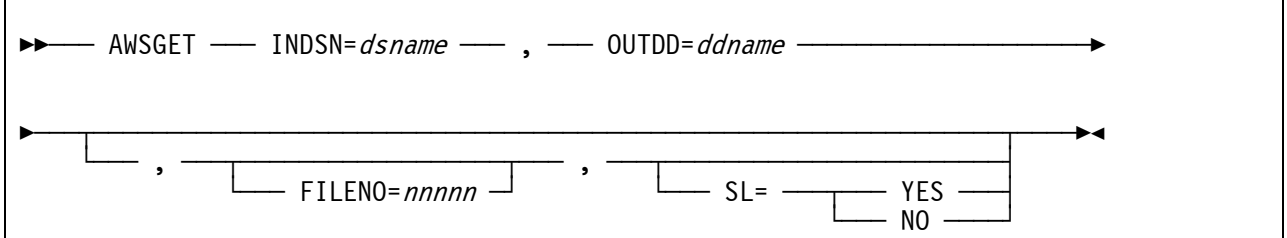
## *8.28.4 AWSGET Control Statement*

The AWSGET control statement will supply values that are necessary to retrieve a dataset from an AWS or HET virtual tape volume. Decompression is invoked automatically when required.

**Descriptive**

AWSGET INDSN=*dsname*,OUTDD=*ddname* [,FILENO=*nnnnn*] [,SL={YES | NO}]

**Diagram**




**INDSN=dsname** This specifies the dataset name (up to 44 bytes) of a dataset stored inside of the AWS or HET virtual tape.

**OUTDD=ddname** This is the DDNAME representing the dataset to receive the data from the virtual tape.

**FILENO=nnnnn** Specifies the file number of the dataset inside of the AWS or HET virtual tape. This may be a standard label file number or an absolute file number depending on the value of the SL= keyword.

**SL=value** If the value is YES then standard label numbers are present, if the value is NO then the file numbers are absolute numbers. This keyword also impacts the meaning of the FILENO= keyword.

## *8.28.5 AWSPUT Control Statement*

The AWSPUT control statement causes a dataset to be copied into the AWS or HET virtual tape file. Multiple AWSPUT statements may be specified. A set of standard labels are produced as each statement is processed.

If necessary the dataset is staged into a temporary dynamically allocated dataset prior to insertion into the virtual tape.

The AWS or HET virtual tape output file may specify any DCB attributes that are meaningful in the users environment:

**RECFM=V** Variable length output, LRECL and BLKSIZE as specified. The records are in a format consistent with that produced by AWSUTIL.

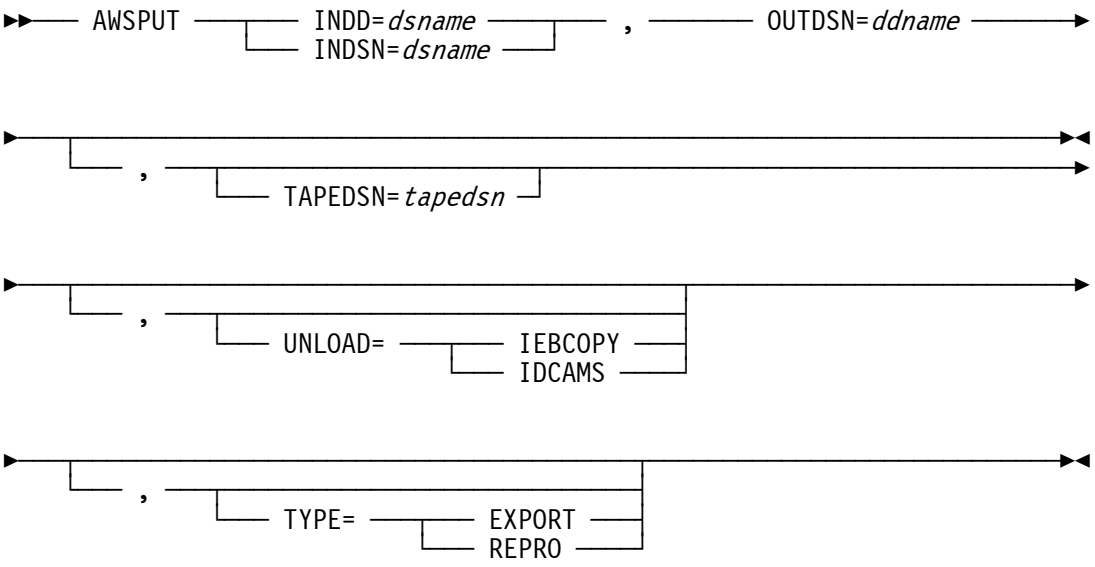
**RECFM=F** Fixed length output, LRECL and BLKSIZE as specified. The records are in a format consistent with that produced by the VTTS family of utilities.

**RECFM=U** Undefined length output, BLKSIZE as specified. The records are written in an aggregated BLKSIZE length block. *Note:* This is the preferred method when the virtual tape is to be transported to other environments such as Hercules.

**Descriptive**

```
AWSPUT {INDD=ddname | INDSN=dsname},OUTDSN=dsname [,TAPEDSN=tapedsn]
       [,UNLOAD={IEBCOPY | IDCAMS}] [,TYPE={EXPORT | REPRO}]
```

**Diagram**




**INDD=ddname** Specifies the statically allocated DDNAME representing the file to be copied and placed into the AWS or HET virtual tape file.

| | |
|---|---|
| **INDSN=dsname** | Specifies the dsname to be dynamically allocated and placed into the AWS or HET virtual tape file. |
| **OUTDSN=dsname** | The dataset name (up to 44 bytes) to be placed into the labels which preceed and follow the file on the virtual tape. |
| **TAPEDSN=tapedsn** | The datasetname (up to 17 bytes) to be placed in the label. |
| **UNLOAD=value** | Selects the utility (IEBCOPY or IDCAMS) to be called to stage the input dataset prior to insertion into the virtual tape. |
| **TYPE=value** | If UNLOAD=IDCAMS is selected then the TYPE value specifies the method (EXPORT or REPRO) to be used to stage the dataset prior to insertion into the virtual tape. |

## *8.28.6 Samples*

**Sample 1:**

Create an AWS virtual tape from several input datasets/input DDs:

```
//VIRTTAPE EXEC PGM=AWSSL
//STEPLIB  DD   DSN=your.loadlib,DISP=SHR
//AWSPRINT DD   SYSOUT=*
//myddnam1 DD   DSN=your.file1,DISP=SHR
//myddnam2 DD   DSN=your.file2,DISP=SHR
//AWSFILE  DD   DSN=volser.aws,DISP=(NEW,CATLG,DELETE),
//           SPACE=(your_space_parms),
//           DCB=(your_DCB_parms)
//AWSCNTL  DD *
AWSVOL  VOLSER=mytape,COMPRESS=0
AWSPUT  INDSN=your.first.dataset
AWSPUT  INDSN=your.second.dataset,UNLOAD=IEBCOPY
AWSPUT  INDD=myddnam1
AWSPUT  INDD=myddnam1,UNLOAD=IEBCOPY
/*
```

**Figure 233: AWSSL JCL (Create AWS virtual tape)**

**Sample 2:**

Create datasets from files on an AWS virtual tape:

```
//VIRTTAPE EXEC PGM=AWSSL
//STEPLIB  DD   DSN=your.loadlib,DISP=SHR
//AWSPRINT DD   SYSOUT=*
//myddnam1 DD   DSN=my.file1,DISP=NEW,CATLG,DELETE),
//           SPACE=(your_space_parms),
//           DCB=(your_DCB_parms)
//myddnam2 DD   DSN=my.file2, DISP=NEW,CATLG,DELETE),
//           SPACE=(your_space_parms),
//           DCB=(your_DCB_parms)
//AWSFILE  DD   DSN=volser.aws,DISP=SHR
//AWSCNTL  DD *
AWSVOL  VOLSER=mytape
AWSGET  OUTDD=myddnam1,INDSN=your.dataset1,FILENO=1,SL=YES
AWSGET  OUTDD=myddnam2,INDSN=your.dataset2,FILENO=5,SL=YES
/*
```

**Figure 234: AWSSL JCL (Create datasets from AWS virtual tape)**

## 8.29 AWSUTIL (AWS format tape file generation utility)

AWSUTIL is written by Brandon Hill and is available on the CBT MVS Utilities Tape (CBTTAPE) at www.cbttape.org (File #477).

### *8.29.1 Function*

AWSUTIL is a comprehensive program for creating AWS format virtual tape files. The program can generate labels and write tape marks. Any dataset that can be accessed by BSAM can be copied to a virtual tape file. Any real tape device can have either a single file or the entire tape copied.

The commands for the utility are read from the SYSIN DD statement and status and errors are written to the SYSPRINT DD statement. The AWS tape image is written to SYSOUT as a VB file. This file is then suitable to be copied with FTP in binary mode to any other system.

Any number of DD statements representing sequential data and tapes can be specified and are read via the appropriate statement in SYSIN.

### *8.29.2 Control Statements*

AWSUTIL supports these SYSIN control statements:

| | |
|---|---|
| **READ ddname** | Perform BSAM read from *ddname* and convert each block to AWS tape format. This works on whatever BSAM will read. The maximum BLKSIZE is 32760. |
| **GET ddname** | Perform QSAM read from *ddname* and convert each block to AWS tape format. This works on whatever QBSAM will read. The maximum BLKSIZE is 32760. |
| **TAPEFILE ddname** | Perform EXCP read from a real tape device and convert each block to AWS TAPE format. Only works on a tape device. Maximum BLKSIZE is 65535. This command stops reading when a tape mark is read. Therefore only the current tape file is processed. If BLP were used only the tape headers would be processed. |
| **TAPEALL ddname** | Perform EXCP read from a real tape device and convert each block to AWS TAPE format. Only works on a tape device. Maximum BLKSIZE is 65535. This command rewinds the tape and reads until an error condition is returned. This effectively copies the entire contents of the tape. BLP is not necessary as a rewind command is sent before the tape is read. Tape marks are duplicated as they are encountered. *NOTE:* Because end of reel reflectors are rarely effective this command can run a tape off the reel on a 3420 type drive. 3480/3490/3590 type drives have better stopping mechanisms. |
| **TAPEMARK** | Write a tape mark to the output file. This is usually used to separate individual files on a tape. Two consecutive tape marks sometimes signals the end of the tape. |
| **VERIFY** | Print a message after each command verifying that the command was executed ok. This merely prints a message. It has no effect on the integrity of the data. Because SYSOUT should be directed to disk there should be no data integrity problems anyway. |
| **ASCII** | All subsequent labels generated are converted to ASCII before they are written. This only affects labels. All data is copied exactly as it is read no matter the setting of this command. |

| | |
|---|---|
| **EBCDIC** | All subsequent labels generated are not to be converted before they are written. The implication is that the data is still in EBCDIC format. This only affects labels. All data is copied exactly as it is read no matter the setting of this command. |
| **VOL1** | These commands write 80 byte labels to the tape image. The format of the labels is controlled by the ASCII and EBCDIC commands. Except for the first 4 bytes of each input line none of the data is checked to make sure it is a valid tape label. It is up to the user to provide correctly formatted labels as desired. Essentially the entire 80 byte input record is written to the AWS TAPE image as is (or converted to ASCII if this flag is set). |
| **HDR1 - HDR2** | See VOL1 above. |
| **EOF1 - EOF2** | See VOL1 above. |
| **EOV1 - EOV2** | See VOL1 above. |
| **UHL1 - UHL8** | See VOL1 above. |
| **UTL1 - UTL8** | See VOL1 above. |

## *8.29.3 Examples*

The following example reads the first dataset from DASD and a second one dataset from tape and writes them to an AWS format dataset.

```
//MAKETAPE EXEC PGM=AWSUTIL
//STEPLIB  DD   DISP=SHR,DSN=my.load.library
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   DISP=(,CATLG),SPACE=(TRK,(15,15)),DSN=my.dsname
//INDATA   DD   DISP=SHR,DSN=my.input.sequential.data
//INTAPE   DD   DISP=OLD,DNS=my.real.tape.file
//SYSIN    DD   *
READ INDATA
TAPEMARK
TAPEFILE INTAPE
TAPEMARK
/*
```

**Figure 235: AWSUTIL utility JCL**

## 8.30 RAWSTAPE (Reverse AWSTAPE utility)

The RAWSTAPE program is written by Jan Jaeger and is available on the CBT MVS Utilities Tape (CBTTAPE) at www.cbttape.org (File #478).

### *8.30.1 Function*

The RAWSTAPE utility converts an AWSTAPE file back to a blocked OS dataset such that the original blocking is restored. As an AWSTAPE file does not contain DCB parameters this utility will not restore the original DCB parameters, these will have to be added manually with a utility such as IEBGENER.

Upload the AWSTAPE file to OS/390 or z/OS in binary format, do not use CR/LF or any other blocking feature when doing the file transfer. The file must be uploaded to a RECFM=U dataset, either preallocated or allocated using IND$FILE. BLKSIZE=4096 is reasonable but any other blocksize will work also.

An AWS tape file can consist of more then 1 file and as such the filenumber of the tape must be specified. This filenumber is equal to the value used when reading a tape with bypass label processing (BLP). Thus, on a labeled tape file 1 will always be the VOL1 record.

Run RAWSTAPE with SYSUT1 pointing to AWS input file and SYSUT2 to the output file, for example copying a standalone dump tape.

### *8.30.2 Examples*

The following example copies an AWS tape format dataset back to a blocked OS dataset:

```
//CONVERT  EXEC PGM=RAWSTAPE,PARM=3
//STEPLIB  DD   DSN=IBMUSER.LOAD,DISP=SHR
//SYSUDUMP DD   SYSOUT=*
//SYSUT1   DD   DISP=SHR,DSN=IBMUSER.SADUMP.AWS
//SYSUT2   DD   DSN=IBMUSER.SADUMP,DISP=(NEW,CATLG),
//            UNIT=SYSALLDA,SPACE=(TRK,1200,RLSE)
```

**Figure 236: RAWSTAPE utility JCL**

After the previous example the IBMUSER.SADUMP contains the correct data but still has invalid DCB parameters in the DSCB, these can be corrected with IEBGENER as following:

```
//SETDCB   EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   DUMMY
//SYSUT1   DD   DUMMY,DCB=(DSORG=PS,RECFM=FBS,LRECL=4160,BLKSIZE=29120)
//SYSUT2   DD   DISP=MOD,DCB=(*.SYSUT1),DSN=IBMUSER.SADUMP
```

**Figure 237: IEBGENER JCL for RAWSTAPE utility**

## 8.31 VTTS (Virtual tape transportation system)

The programs presented in this section are part of the VTTS (Virtual tape transportation system) and are copyrighted and owned by Sam Golob, however the programs on CBTTAPE File #533 can be used freely. The CBT MVS Utilities Tape (CBTTAPE) can be found at www.cbttape.org.

File #533 of the CBT tape contains the following programs:

- VTT2DISK
- VTT2TAPE
- VTT2CNVU
- VTT2FK2T
- VTT2T2FK

### *8.31.1 Function*

The VTT2TAPE program converts AWS-format tape files to real tapes, while the VTT2DISK program creates an AWS-format virtual tape file from a real tape. The VTT2CNVU program converts a VB-format AWS-format tape (such as the one produced by the AWSUTIL program) to FB-80 format on MVS so that VTT2TAPE can be used subsequently to convert the data to a real tape.

The VTT2T2FK program is similar to VTT2DISK except that a real tape is converted to a FAKETAPE (tm) format tape (see below). VTT2FK2T is similar to VTT2TAPE except that a FAKETAPE tape image, folded over on MVS into FB-80 format, is converted into a real tape. Unless otherwise mentioned AWS-format virtual tape files on an MVS system have been folded over into fixed blocked 80-byte record format. AWS-format files on other systems are just long strings of data.

The FAKETAPE file format is a published interface of Fundamental Software Inc. Fundamental Software reserves the right to change the format at any time. FAKETAPE and FLEX-ES are registered trademarks of Fundamental Software Inc.

The AWS Tape format is produced by IBM's P/390 and Hercules systems to create disk files which an MVS system reads and writes as if they were actual tapes. Another name for these disk files is "virtual tapes", as the MVS system looking at these files "thinks" that they really are tapes.

There are programs like AWSUTIL which can read real tapes and create virtual tapes from them. These virtual tapes can be loaded on a Hercules system somewhere and be read as a tape.

VTT2DISK performs this function also, by reading a real tape and producing an FB-80 folded AWS-format tape as an MVS disk file. While reading the tape the VTT2DISK program produces a lot of statistics about the tape. The advantage of the VTT2DISK program is that it can be run on mainframe-based MVS systems that cannot normally produce AWS-format virtual tapes.

Another issue is encountered when transferring data in the other direction. Suppose you have an MVS system that cannot read a tape in AWS format. Most MVS systems which are not P/390 or Hercules machines are in this position. For these you have to convert an AWS virtual tape disk file to some format that the MVS system can convert into a REAL tape. That is the purpose of the VTT2TAPE program.

### *8.31.2 Control Statements*

The VTT2DISK program allows the following keywords via the SYSIN DD Statement:

| | |
|---|---|
| **CHUNKSIZE=nnnn** | Specifies the chunksize. Any chunksize up to the limit of 65535 bytes is allowed. If not specified the default is 65535 bytes. |
| **NEWVOL=volser** | This keyword changes the VOLSER on VOL1 label to *volser*. |
| **READ** | The READ keyword produces a read only run, no AWSOUT is created. |

**IDRCOFF** This keyword turns off the "data is compressed" indicators in the tape labels.

### *8.31.3 Examples*

**Example 1:**

Read a tape and write it to an AWS format file.

```
//*******************************************************************
//* VTT2DISK - COPY TAPE TO DISK (AWS TAPE FORMAT)                  *
//*******************************************************************
//AWSWRITE EXEC   PGM=VTT2DISK
//STEPLIB  DD     DSN=your.loadlib,DISP=SHR
//AWSOUT   DD     DSN=your.aws.file,DISP=(NEW,CATLG),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=27920),
//             SPACE=(CYL,(750,750),RLSE)
//TAPIN     DD    DISP=OLD,UNIT=3490,VOL=SER=xxxxxx,
//             LABEL=(1,BLP,EXPDT=98000)
//SYSPRINT  DD  SYSOUT=*
//SYSFILES  DD  SYSOUT=*
//SYSTRACE  DD  SYSOUT=*
//SYSMOVED  DD  SYSOUT=*
//SYSUDUMP  DD  SYSOUT=*
```

**Figure 238: VTT2DISK utility JCL.**

**Example 2:**

Read an AWS format file and write it to a tape.

```
//*******************************************************************
//* VTT2TAPE - COPY DISK (AWS TAPE FORMAT) TO REAL TAPE             *
//*******************************************************************
//AWSREAD  EXEC   PGM=VTT2TAPE
//STEPLIB  DD     DSN=your.loadlib,DISP=SHR
//AWSIN    DD     DSN=your.aws.file,DISP=SHR
//TAPOUT   DD     DISP=OLD,UNIT=3490,VOL=SER=xxxxxx,
//             LABEL=(1,BLP,EXPDT=98000)
//SYSPRINT DD     SYSOUT=*
//SYSFILES DD     SYSOUT=*
//SYSTRACE DD     SYSOUT=*
//SYSMOVED DD     SYSOUT=*
//SYSUDUMP DD     SYSOUT=*
```

**Figure 239: VTT2TAPE utility JCL**

**Example 3:**

Convert an AWSUTIL file (VB) to standard AWS format (FB).

```
//*******************************************************************
//* VTT2CNVU - CONVERT AWSUTIL FILE (VB) TO STANDARD AWS FORMAT (FB)  *
//*******************************************************************
//AWSCNVT  EXEC   PGM=VTT2CNVU
//STEPLIB  DD     DSN=your.loadlib,DISP=SHR
//AWSVBIN  DD     DSN=your.vbaws.file,DISP=SHR
//AWSFBOUT DD     DSN=your.fbaws.file,DISP=(NEW,CATLG),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=27920),
//             SPACE=(CYL,(300,50),RLSE)
//SYSPRINT DD     SYSOUT=*
//SYSFILES DD     SYSOUT=*
//SYSTRACE DD     SYSOUT=*
//SYSMOVED DD     SYSOUT=*
//SYSUDUMP DD     SYSOUT=*
```

**Figure 240: VTT2CNVU utility JCL**

# 9. Shared Device Support

## 9.1 Basics

Shared Device Support (see also "General Information" manual) allows multiple Hercules instances to share devices. The device will be local to one Hercules instance and remote to all other Hercules instances. The local instance is the server for that device and the remote instances are the clients. It is possible that each instance acts as both a client and a server. If a local instance declares a device as remote on another instance and on this remote instance the device is defined again as remote on a third instance, then the original instance will have to hop through this second instance to get to the real device.

It is not necessary to IPL an operating system on the device server. Any number of Hercules instances can act as a server in a "HERCPLEX".

When "SHRDPORT" is specified in the Hercules configuration the thread "shared_server" is started at the end of Hercules initialization. If Shared Device Support is requested on a device statement then the Hercules instances cannot initialize these devices until the server is started on each system. In this case the device trying to access a server gets the 'connecting' bit set on in the DEVBLK and the device still needs to initialize. After the shared server is started a thread is attached for each device that is connecting to complete the connection (the device init handler).

## 9.2 Caching

Cached records (i.e. CKD tracks or FBA blocks) are kept independently on both the client and server sides. Whenever the client issues a START request to initiate a channel program, the server will return a list of records to purge from the clients cache. These will have been updated by other clients since the last START request. If the list is too large the server will indicate that the client should purge all records for the device.

## 9.3 Compression

Data that would normally be transferred uncompressed between the client and the host can optionally be compressed by specifiying the "COMP=n" keyword on the device configuration statement (see below) or on the attach command. The value n of the "COMP=n" keyword is the zlib compression parameter which must be a number between 1 and 9. A value closer to 1 means less compression but less processor time to perform the compression. A value closer to 9 means the data is compressed more but also more processor time is required to compress the data.

If the server is on localhost then you should not specify compression. Otherwise you are just stealing processor time from Hercules to facilitate compression/decompression. If the server is on a local network then a low value such as 1, 2 or 3 is recommended. There is a tradeoff curve, attempting to trade CPU cycles for network traffic to derive an optimal throughput.

If the devices on the server are compressed devices (i.e. CCKD or CFBA) then the records (track images or block groups) may be transferred compressed regardless of the "COMP=n" settings. This depends on whether the client supports the compression type (zlib or bzip2) of the record on the server and whether the record is actually compressed in the server cache.

An example may help to explain this: Suppose on the client that you execute one or more channel programs to read a record on a CKD track, update a record on the same track, and then read another (or the same) record on the track. For the first read the server will read the track image and pass it to the client as it was originally compressed in the file. To update a portion of the track image the server must uncompress the track image so data in it can be updated. When the client next reads from the track image the track image is uncompressed.

Specifiying "COMP=n" means that uncompressed data sent to the client will be compressed. If the data to be sent to the client is already compressed then the data is sent as is unless the client has indicated that it does not support that compression algorithm.

## 9.4 Usage of Shared Devices

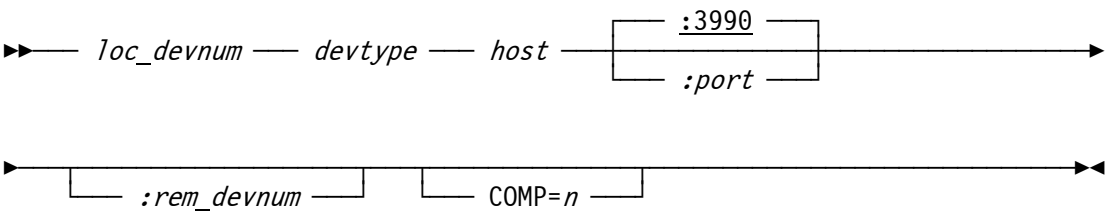
To use a device on a remote Hercules instance, instead of specifying a file name on the device statement, an IP address or a DNS name is specified.

### *9.4.1 Syntax*

**Descriptive**

*loc_devnum devtype host*[*:port*] [*:rem_devnum*] [COMP=*n*]

**Diagram**




### *9.4.2 Parameter*

**loc_devnum** This specifies the device address on the local Hercules instance.

**devtype** This is the device type.

**host** This specifies the host name or the IP address of the system where the Shared Device Server is running.

**port** The port number on which the Shared Device Server is listening. If the port number is omitted then the default port (3990) is used.

**rem_devnum** This is the device address of the device on the remote Hercules instance. If the remote device address is omitted then the default is the current device number on the local system.

**COMP=n** This keyword requests that the data has to be transferred compressed between the client and the server. The argument *n* specifies the compression level (1-9). A value closer to 1 means less compression but also less processor time to perform the compression. A value closer to 9 means the data is compressed more but also more processor time is required to compress the data.

### *9.4.3 Examples*

**Example 1:**

There is a shared device server on the local host listening on port 3990 and we want to use its 0100 device as our 0100 device. The device statement will look like this:

```
0100 3390 localhost:3990:0100
```

Because the default port number is actually 3990 and the default remote device address is the same as the local device number the above statement (providing we do not have actually a file named 'localhost' ) could be shortened as follows:

```
0100 3390 localhost
```

**Example 2:**

Device sharing can also be split between multiple instances. For example, suppose the following device definitions and system parameter for instance A:

```
SHRDPORT 3990
0100 3390 localhost:3991
0101 3390 mvscat.dasd
```

And for instance B we have the following device definitions and system parameter:

```
SHRDPORT 3991
0100 3390 mvsres.dasd
0101 3390 localhost
```

In this case each instance acts as both a client and a server. Both instances of Hercules are running on the same machine.

The above examples may be clearer if we specify also all the default values. To show this the same configuration is used but in this case the Hercules instances are running on separate physical machines:

Hercules instance A (machine IP 192.168.200.1):

```
SHRDPORT 3990
0100 3390 192.168.200.2:3991:0100   # Remote on 192.168.200.2 (mvsres.dasd)
0101 3390 mvscat.dasd               # Local on  192.168.200.1 (mvscat.dasd)
```

Hercules instance B (machine IP 192.168.200.2):

```
SHRDPORT 3991
0100 3390 mvsres.dasd               # Local  on 192.168.200.2 (mvsres.dasd)
0101 3390 192.168.200.1:3990:0101   # Remote on 192.168.200.1 (mvscat.dasd)
```

# 10. Hercules 3270 Logo

## 10.1 Function

The Hercules 3270 logo is the initial welcome screen that is presented when a 3270 terminal connects to a Hercules 3270 device. While in previous releases of Hercules this logo screen was hardcoded it can now be customized.

The customized logo is stored in a plain text file which contains positioning orders, attributes and supports variable substitutions. Each line in the text file represents either an order statement or a plain text line. If no logo file is specified then Hercules uses a built-in default logo. Upon startup Hercules will first look in the current directory for a file called "herclogo.txt". The name of the logo file can also be specified as a startup option by using the "-b" flag. The file may also be specified using the "LOGOFILE" configuration statement or can be changed during Hercules operation using the "HERCLOGO" console command.

## 10.2 Orders

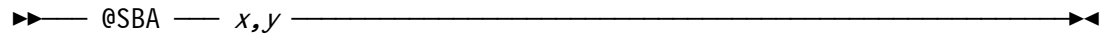
Order commands are used to set the current buffer position, set the highlight and/or protected attributes, to force a skip to a new line and to specifiy text alignment.

### *10.2.1 Set Buffer Address*

**Descriptive**

`@SBA x,y`

**Diagram**



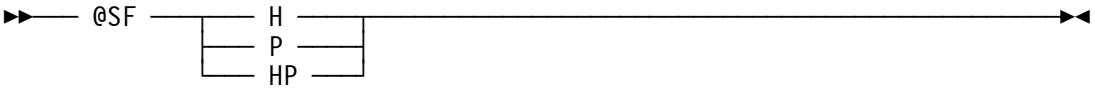

The Set Buffer Address order sets the current buffer position to row x and column y. Row and column both begin counting with zero.

### *10.2.2 Set Field*

**Descriptive**

`@SF {H | P | HP}`

**Diagram**



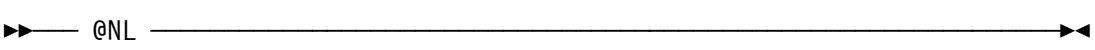

The Set Field order sets the highlight ("H") and/or protected ("P") attribute.

### *10.2.3 New Line*

**Descriptive**

```
@NL
```

**Diagram**



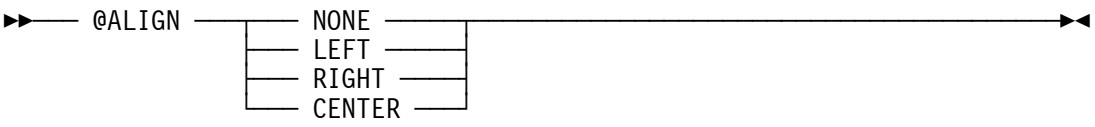

The New Line order forces a skip to a new line.

### *10.2.4 Align*

**Descriptive**

```
@ALIGN {NONE | LEFT | RIGHT | CENTER}
```

**Diagram**




The Align order specifies the text alignment relative to the left and reight borders of the terminal. When the alignment is other than "NONE" a new line is automatically inserted after each line of text. If the alignment is "NONE" then the text will be written without skipping to the next line.

## 10.3 Variables

It is possible to imbed substitution variables in the outgoing text. Substitution is indicated by enclosing the variable name between "$(" and ")". The Hercules version as an example can be specified by the following string: $(VERSION)

The following variables are defined. It is also possible to specify environment variable names.

- $(VERSION)
  The Hercules version.
- $(HOSTNAME)
  The host name, on which Hercules is running.
- $(HOSTOS)
  The host operating system.
- $(HOSTOSREL)
  The release of the host operating system.
- $(HOSTOSVER)
  The version of the host operating system.
- $(HOSTARCH)
  The host architecture.
- $(HOSTNUMCPUS)
  The number of host CPUs. UP (Uniprocessor for one CPU), or MP=n (Multiprocessor for more than one CPUs).
- $(LPARNAME)
  The LPAR name specified in the configuration file.
- $(CSS)
  The logical channel subsystem set or channel set for the terminal.
- $(SUBCHAN)
  The subchannel number for the terminal.
- $(CCUU), $(ccuu), $(CUU), $(cuu)
  Various forms of the device number of the terminal.

## 10.4 Sample

The file "herclogo.txt" is provided in the Hercules distribution as a sample template. This sample reflects the contents of the built-in logo.

```
@ALIGN NONE
@SBA 0,0
@SF P
Hercules Version  :
@SF HP
$(VERSION)
@NL
@SF P
Host name         :
```

```
@SF HP
$(HOSTNAME)
@NL
@SF P
Host OS           :
@SF HP
$(HOSTOS)-$(HOSTOSREL) $(HOSTOSVER)
@NL
@SF P
Host Architecture :
@SF HP
$(HOSTARCH)
@NL
@SF P
Processors        :
@SF HP
$(HOSTNUMCPUS)
@NL
@SF P
Chanl Subsys      :
@SF HP
$(CSS)
@NL
@SF P
Device number     :
@SF HP
$(CCUU)
@NL
@SF P
Subchannel        :
@SF HP
$(SUBCHAN)
@SF P
@ALIGN LEFT


           HHH          HHH   The S/370, ESA/390 and z/Architecture
           HHH          HHH                 Emulator
           HHH          HHH
           HHH          HHH  EEEE RRR   CCC U  U L    EEEE  SSS
           HHHHHHHHHHHHHHHHH  E    R  R C    U  U L    E    S
           HHHHHHHHHHHHHHHHH  EEE  RRR  C    U  U L    EEE   SS
           HHHHHHHHHHHHHHHHH  E    R R  C    U  U L    E       S
           HHH          HHH  EEEE R  R  CCC  UU  LLLL EEEE SSS
           HHH          HHH
           HHH          HHH
           HHH          HHH     I can't believe it's not a MAINFRAME

           Copyright (c) 1999-2010 Roger Bowler, Jan Jaeger, and others
```

**Figure 241: Logo File**

This above sample file results in the following Hercules Logo screen:

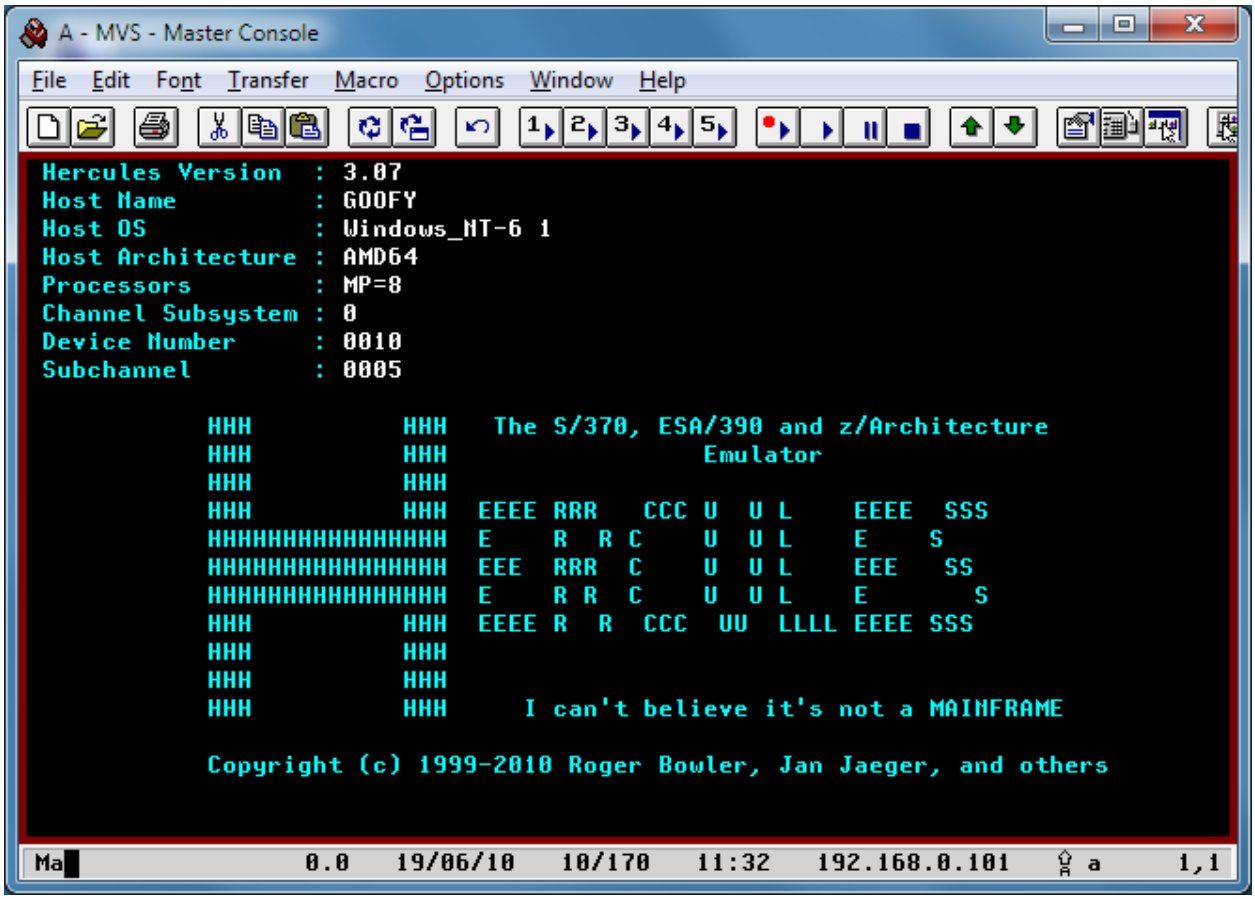



**Figure 242: Logo Screen**

# 11. Starting the Hercules Emulator

## 11.1 Starting Hercules in Native Mode

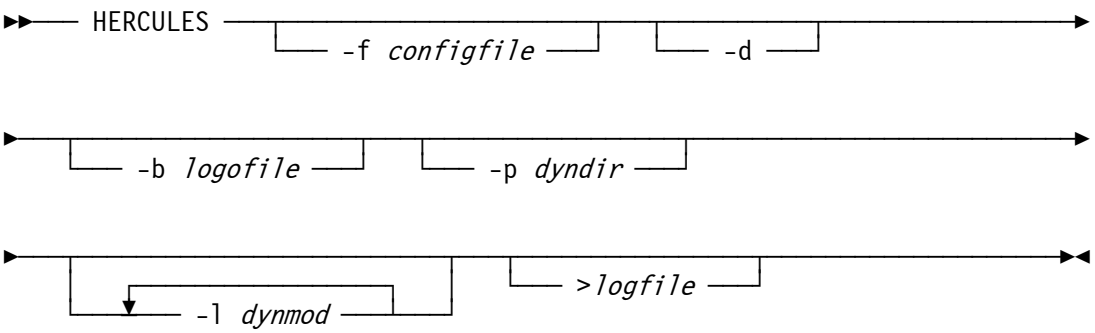
Hercules can be manually started from a Windows Command Prompt or can be started by calling a batch file (the preferred way). In both cases the syntax is the same.

### *11.1.1 Syntax*

**Descriptive**

HERCULES [-f *configfile*] [-d] [-b *logofile*] [-p *dyndir*] [[-l *dynmod*] ... ]
[>*logfile*]

**Diagram**




### *11.1.2 Parameter*

**-f configfile** This is the name of the configuration file. If the filename is not specified the default is HERCULES.CNF. The name of the default configuration file may be overriden via the 'HERCULES_CNF' environment variable.

**-d** Specifies that Hercules is to be run in “daemon” mode, wherein it runs invisibly with no attached console.

**-b logofile** Specifies the name of a customized logo text file in the current Hercules directory (for details on how to customize a logofile see chapter 10).

**-p dyndir** This is the directory from which dynamic modules are to be loaded. The default depends on the host platform on which Hercules is being run. This option overrides the default.

**-l dynmod** This is the name of an additional dynamic load module to be loaded at startup. More than one additional module may be specified, although each must be preceded with the *“-l”* option.

**>logfile** This is an optional log file which will receive a copy of all messages displayed on the control panel.

### *11.1.3 Examples*

**Example 1:**

Start Hercules with the configuration file “D:\MVS\CONF\MVS38J.CONF” and write all messages to the log file “D:\MVS\LOGS\HERCULES.LOG”.

```
HERCULES -f D:\MVS\CONF\MVS38J.CONF >D:\MVS\LOGS\HERCULES.LOG
```

**Example 2:**

Start Hercules with the configuration file “D:\S390\CONF\zLINUX.CNF” and with a logofile called “zLINUX_Logo.txt”.

```
HERCULES -f D:\S390\CONF\zLINUX.CNF -b zLINUX_Logo.txt
```

## 11.2 Starting Hercules with the Windows GUI

The Windows GUI can be started by just clicking on the GUI icon. Then all necessary settings required to start Hercules can be specified using the GUI itself.

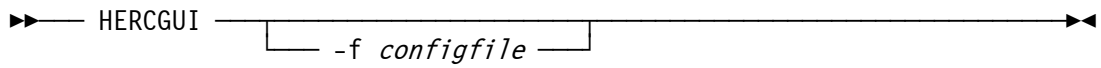
Alternatively the Windows GUI may also be manually started from a Windows Command Prompt or can be started by calling a batch file (which is the preferred way of these two options). In both cases the syntax is the same.

### *11.2.1 Syntax*

**Descriptive**

HERCGUI [-f *configfile*]

**Diagram**




### *11.2.2 Parameter*

**-f configfile** This is the name of a configuration file. If the HercGUI is started with the name of a configuration file then the HercGUI will automatically “power on” Hercules once the HercGUI itself is started. This is called the “Auto-Power-On” feature. When using this feature in conjunction with a Hercules “.RC” (run commands) file it is possible to totally automate Hercules startup and IPL.

### *11.2.3 Examples*

**Example 1:**

Start Hercules using the HercGUI with the “Auto-Power-On” feature using the configuration file “D:\MVS\CONF\MVS38J.CONF”.

```
HERCGUI -f D:\MVS\CONF\MVS38J.CONF
```

**Example 2:**

Start the HercGUI without a configuration file.

```
HERCGUI
```

## 11.3 Starting Hercules with the Hercules Studio

The Hercules Studio can be started by navigating to the 'Applications -> System Tools -> Hercules Studio' menu entry and selecting it from there. Then all necessary settings required to start Hercules can be specified using the Hercules Studio itself.

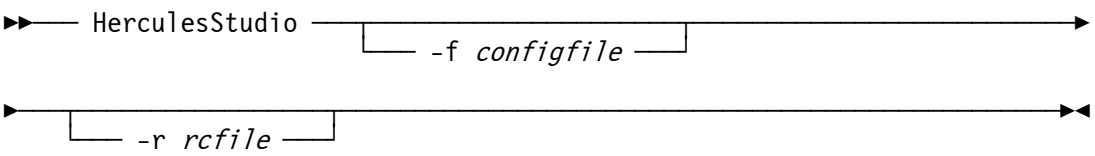
Alternatively the Hercules Studio may also be manually started from a command shell or can be started by calling a script file. In both cases the syntax is the same.

### *11.3.1 Syntax*

**Descriptive**

HerculesStudio [-f *configfile*] [-r *rcfile*]

**Diagram**




### *11.3.2 Parameter*

**-f configfile** This is the name of a configuration file. If the Hercules Studio is started with the name of a configuration file then it will automatically “power on” Hercules once the Hercules Studio itself is started. This is called the “Auto-Power-On” feature. When using this feature in conjunction with a Hercules “.RC” (run-commands) file it is possible to totally automate Hercules startup and IPL.

**-f rcfile** This is the name of an optional RC (run-commands) file. This file contains Hercules console commands that are issued exactly as if the commands were entered from the Hercules system console.

### *11.3.3 Examples*

**Example 1:**

Start Hercules Studio using the configuration file “mvs38j.cnf” and the run-commands file "autoipl.rc".

```
HerculesStudio -f mvs38j.cnf –r autoipl.rc
```

# 12. The Run-Commands File (hercules.rc)

## 12.1 Function

Hercules supports the ability to have console commands automatically executed at startup via the 'run-commands' file. If the run-commands file is found to exist when Hercules starts then each line of the file is read and interpreted as a console command, exactly as if the command were entered from the HMC system console.

The default filename for the run-commands file is "hercules.rc". The default name may be overridden by setting the "HERCULES_RC" environment variable.

Except for the 'pause' command (see below), each command read from the run-commands file is logged to the console preceded by a “>"' (greater-than sign) character. Thus you can easily distinguish between console commands entered from the keyboard from those entered via the run-commands file.

Comment lines starting with '#' are treated as "silent comments" and are thus not logged to the console whereas comment lines starting with '*' are treated as "loud comments" and *will* be logged.

## 12.2 Run-Commands File Statements

There is a number of different types of commands that can be specified within a run-commands file. These commands are the following:

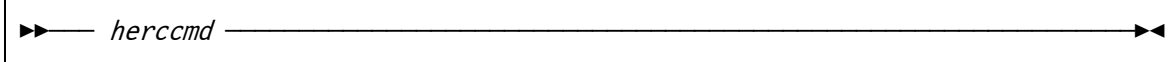
- “*herccmd*” (any valid Hercules console command)
- PAUSE (delay the processing of the next command)
- # (issue silent comment)
- * (issue loud comment)

### *12.2.1 Hercules Console Commands*

**Descriptive**

*herccmd*

**Diagram**




In the run-commands file any valid Hercules console command (“*herccmd*”) can be specified. This includes the ‘sh’ (shell) command. Each command that is read from the run-commands file is logged to the console preceded by a ‘>" (“greater-than” sign) character. This is to distinguish between console commands entered manually from the keyboard from thosecommands entered via the run-commands file.

Examples:

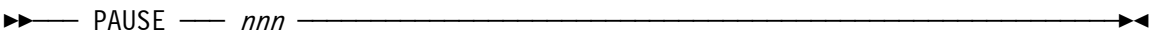
```
MAXRATES MIDNIGHT
IPL 0148
```

### *12.2.2 PAUSE*

**Descriptive**

PAUSE *nnn*

**Diagram**




A pause command is supported in order to introduce a brief delay before reading and processing the next line in the file. The value *nnn* can be any number from 1 to 999 and specifies the number of seconds to delay before reading the next line.

Examples:

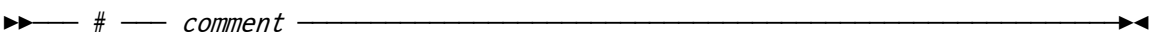
```
PAUSE 2
```

### *12.2.3 # (Silent Comment)*

**Descriptive**

# [*comment*]

**Diagram**




Lines starting with '#' are treated as "silent comments". They are intended as comments just for the run-commands file itself and are thus not logged to the console. That means they are are not visible when Hercules is started and the run-commands file is executed.

Examples:

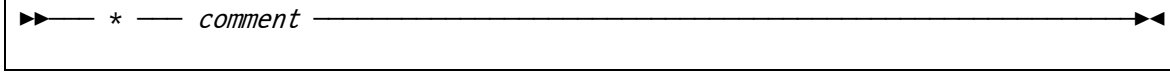
```
# ================================================
# This comment block will not be logged to the console
# ================================================
```

### *12.2.4 * (Loud Comment)*

**Descriptive**

`* [comment]`

**Diagram**




Lines starting with '*' are treated as "loud comments". They will be logged to the console and thus are visible when the run-commands file is executed.

Examples:

```
* ================================================
* This comment block will be logged to the console
* ================================================
```

## 12.3 Automating Hercules Startup

Creative use of the run-commands file can completely automate Hercules startup. The next figure shows a sample run-commands file that does the following:

- “pause 1” – Wait one second before processing the first Hercules command. (1)
- “traceopt traditional” – Specify the traditional trace mode (registers are displayed following the instruction). (2)
- “help” – Display all valid Hercules commands. (3)
- “maxrates midnight” – Set the MAXRATES reporting interval to midnight to have the MIPS and I/O statistics issued date aligned. (4)
- “sh startgui "D:\MVS\TERMINALS.BAT" – Execute the named batch file in the shell to start a couple of tn3270 sessions. See details below. (5)
- “pause 3” – Wait three seconds until all tn3270 sessions have been successfully connected. (6)

- “ipl 0148” – Finally IPL MVS 3.8J from device address 148. (7)

```
# ================================================================================
# Hercules RUN-COMMANDS (RC) File
# ================================================================================

# --------------------------------------------------------------------------------
# Wait 1 second before issuing Hercules commands                           (1)
# --------------------------------------------------------------------------------
pause 1

# --------------------------------------------------------------------------------
# Specify the traditional trace mode                                       (2)
# --------------------------------------------------------------------------------
traceopt traditional

# --------------------------------------------------------------------------------
# Display all valid Hercules panel commands                                (3)
# --------------------------------------------------------------------------------
help

# --------------------------------------------------------------------------------
# Display and reset maxrates values                                        (4)
# --------------------------------------------------------------------------------
maxrates midnight

# --------------------------------------------------------------------------------
# Start tn3270 sessions                                                    (5)
# --------------------------------------------------------------------------------
sh startgui "D:\MVS\TERMINALS.BAT"

# --------------------------------------------------------------------------------
# Wait 3 second before IPL’ing MVS                                         (6)
# --------------------------------------------------------------------------------
pause 3

# --------------------------------------------------------------------------------
# IPL MVS 3.8J                                                             (7)
# --------------------------------------------------------------------------------
ipl 0148

# ================================================================================
# (RC EOF)
# ================================================================================
```

**Figure 243: Run-commands (.rc) file**

With the run-commands file shown above the Hercules startup is fully automated. After starting Hercules there is no manual intervention required until MVS is up and running.

The next figure shows the batch file that is used in the run-commands file (step 5) to start the tn3270 sessions.

```
@ECHO OFF

REM ************************************************************************************************
REM Start master console and wait for connection to Hercules
REM ************************************************************************************************
START /D "C:\Program Files (x86)\TomBrennanSoftware\VistaTN3270" VistaTN3270.EXE MVS38J_MST.SES
CHOICE /t 1 /d y

REM ************************************************************************************************
REM Start operator console and wait for connection to Hercules
REM ************************************************************************************************
START /D "C:\Program Files (x86)\TomBrennanSoftware\VistaTN3270" VistaTN3270.EXE MVS38J_OPR.SES
CHOICE /t 1 /d y

REM ************************************************************************************************
REM Start TSO terminals and wait for connection to Hercules
REM ************************************************************************************************
START /D "C:\Program Files (x86)\TomBrennanSoftware\VistaTN3270" VistaTN3270.EXE MVS38J_TSO.SES
CHOICE /t 1 /d y
START /D "C:\Program Files (x86)\TomBrennanSoftware\VistaTN3270" VistaTN3270.EXE MVS38J_TSO.SES

EXIT
```

**Figure 244: Batch file to start tn3270 sessions**

The CHOICE command used in the batch file is to issue a delay before trying to connect the next tn3270 session.

The “/t” parameter specifies the number of seconds to pause before a default choice is made. Option “/d” specifies the default choice that is issued after the timeout has expired. The character given as argument must be in the set of choices specified by the “/c” option (defaults are ‘Y’ and ‘N’).

# 13. The “Hercules Automatic Operator” (HAO) Facility

## 13.1 HAO Introduction

The Hercules Automatic Operator (HAO) feature is a facility which can automatically issue panel commands in response to specific messages appearing on the Hercules console.

To use the Hercules Automatic Operator facility, you first define a rule consisting of a target and the associated command. The target is a regular expression pattern used to match against the text of the various messages that Hercules issues as it runs. Whenever a match is found, the rule “fires” and ist associated command is automatically issued.

The Hercules Automatic Operator facility only operates on messages issued to the Hercules console. These messages may originate from Hercules itself or from the guest operating system via the SCP SYSCONS interface or via the integrated console printer-keyboard (3215-C or 1052-C). HAO cannot intercept messages issued by the guest operating system to its own terminals.

## 13.2 Defining HAO Rules

To define a HAO rule, enter the command

```
HAO TGT target
```

to define the rule’s target match pattern, followed by the command

```
HAO CMD command
```

to define the rule’s associated panel command.

The target is a regular expression as defined by your host platform. When running on Linux, Hercules uses POSIX Extended Regular Expression syntax. On a Windows platform, regular expression support is provided by Perl Compatible Regular Expression (PCRE). The HAO facility can only be used if regular expression support was included in Hercules at build time.

The associated *command* is whatever valid Hercules panel command you wish to issue in response to a message being issued that matches the given *target* pattern.

## 13.3 Deleting HAO Rules

To delete a fully or partially defined HAO rule, first use the the following command to get a list of all of the defined (or partially defined) rules

```
HAO LIST [nnn]
```

Where *nnn* is the (optional) number of an existing rule. This gives you the list of all rules with the specified identifier or lists the rule with identifier ‘*nnn*’. Then use the next command to delete the specific rule identified by the identifier ‘*nnn*’

```
HAO DEL nnn
```

To every rule there is a number assigned as the rule is defined. The rules then are subsequently identified by their numeric value.

It is also possible to delete all defined or partially defined rules by issuing the following command

```
HAO CLEAR
```

## 13.4 Substituting Substrings

The command may contain special variables ($1, $2, $3, etc.) which will be replaced by the values of "capturing groups" in the match pattern. A capturing group is a part of the regular expression enclosed in parenthesis which is matched with text in the target message. In this way commands may be constructed which contain substrings extracted from the message which triggered the command.

The following special variables are recognized:

- $1 to $9 – the text which matched the first to nineth capturing group in the target regular expression.
- $` - the text preceeding the regular expression match.
- $' – the text following the regular expression match.
- $$ - replaced by a single dollar sign.

Note that the substitution of a $*n* variable does not occur if there are fewer than *n* capturing groups in the regular expression.

## 13.5 Limitations

The current implementation limits the total number of defined rules to 64. This limit may be raised by increasing the value of the HAO_MAXRULE constant in module "hao.c" and rebuilding Hercules.

All defined rules are checked for a match each time Hercules issues a message. There is no way to specify "stop processing subsequent rules". If a message is issued that matches two or more rules, each associated command is then issued in sequence.

## 13.6 Examples

**Example 1:**

Issue the command "i 0700" in response to the message:

```
HHCTE014I 3270 device 001F client 127.0.0.1 connection reset
```

The following HAO commands define the desired rule:

```
HAO TGT HHCTE014I 3270 device ([0-9A-F]{3,4})
HAO CMD i $1
```

**Example 2:**

Use the dot matrix display of a 3480 tape unit to implement an automatic tape library (please note that the first HAO command has been split over two lines to fit on the page):

```
HAO TGT HHCTA010I ([0-9A-F]{4}): Now displays:
        (?:".{8}" / )?"M([A-Z0-9]{1,6})\s*S"
HAO CMD devinit $1 D:\MVSTAPES\$2.AWSTAPE
```

# 14. Submitting Jobs via the Socket Reader

## 14.1 Socket Reader Basics

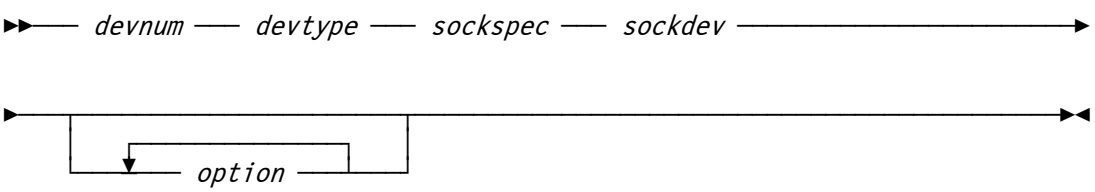
The “sockdev” option allows you to submit cards directly to a Hercules card reader from outside of Hercules. The card reader must be defined with the “sockdev” keyword and either a TCP/IP port number or the name of a Unix Domain Socket. Then whenever you want to submit a card deck to that particular card reader, you use an external program to connect to the socket and transmit the cards to the reader. Socket readers were implemented by Fish, based upon code originally contributed by Malcolm Beattie.

Socket readers are defined in the Hercules configuration like this:

**Descriptive**

*devnum devtype sockspec sockdev* [*option* [*option* ... ]]

**Diagram**




The socket specification *sockspec* can take any of the following formats:

| | |
|---|---|
| **ipaddr:port** | The reader listens on the specific IP address and port number. *ipaddr* must be the IP address of an interface on the local system. For example, “127.0.0.1:1234” is used to accept only jobs submitted locally via the loopback interface. |
| **hostname:port** | Similar to the previous example, where *hostname* must resolve to an IP address belonging to the local system. Example: “localhost:1234”. |
| **port** | The reader listens on the specified port number and accepts jobs submitted from any IP address defined on the local system. Example: “1234”. |
| **sockpath/sockname** | The reader listens on the specified Unix Domain Socket. Example: “/tmp/hercrdr.00C” |

**Examples:**

```
000A   2501   127.0.0.1:2501 SOCKDEV ASCII TRUNC EOF
000C   3505   localhost:1234 SOCKDEV ASCII TRUNC EOF
0012   3505   3505 SOCKDEV ASCII TRUNC EOF
0014   2501   /tmp/hercrdr.014 SOCKDEV ASCII TRUNC EOF
```

## 14.2 Submitting Jobs from Windows

The “HercRdr” program is distributed as part of Fish’s GUI package and allows you to send jobs to a socket reader via TCP/IP. Simply enter “HercRdr” from the command line (i.e. from a “Command Prompt” window if you are using Windows) to submit your file(s).

The following screen print shows the help information from the HercRdr program that is displayed whenever the program is called without parameters:

```
D:\HERCULES>hercrdr
Submits card file(s) to a Hercules card reader bound to a given socket:port.

Format:

  HERCRDR [-nnn] [host:port] file [file...]

Where:

  -nnn        timeout value in seconds (1-999; default is 3)
  host:port   sock_spec of target reader (if not specified,
              value of HERCRDR environ var is used, if set)
  file        file(s) to be submitted

Examples:

  HERCRDR localhost:1234 fileone.txt filetwo.txt
  set HERCRDR=localhost:1234
  HERCRDR file3.txt file4.txt
  HERCRDR override:5678 filefive.txt
  HERCRDR 192.168.0.1:5678 666.txt 777.txt 888.txt 999.txt

Returns:

 -1    unclassified error
  0    file(s) successfully submitted
  1    no route to host (bad sock_spec or connection refused)
  2    timeout value exceeded while trying to connect
  3    transmission error (e.g. connection prematurely closed)
  4    file not found (or other file error)
```

**Figure 245: HercRdr Help Screen**

### *14.2.1 How to submit jobs directly from SPF/PC*

If you are lucky enough to have a copy of the excellent editor SPF/PC Version4 or SPF/Pro (produced by CTC but unfotunately no longer available), then you can submit jobs directly from your edit session. The SUB command can be implemented by means of a REXX macro such as the following one, provided by Volker Bandke:

```
/* +----------------------------- REXX -----------------------------+ */
/*                                                                    */
/*      Name: D:\APPS\SPFPRO\REXX\USER\SUB.SPF                        */
/*                                                                    */
/*      Type: SPF edit macro                                          */
/*                                                                    */
/*      Desc: submit JCL to MVS 3.8                                   */
/*                                                                    */
/*      Creation date: 24 Aug 1999, creation time: 18:49:40           */
/*                                                                    */
/*      Author: (c) Volker Bandke                                     */
/*                                                                    */
/* +----------------------------------------------------------------+ */
'isredit macro (p1 p2 p3 p4 p5 p6 p7 p8 p9)'
"ISREDIT (member) = MEMBER"
"ISPEXEC CONTROL ERRORS CANCEL"
parse upper var member file '.' ext
do
'ISREDIT REPLACE' $$$$$$$$.SPF '.ZF .ZL'
ADDRESS "CMD" "HERCRDR 192.168.1.102:3505 $$$$$$$$.SPF"
zedsmsg = 'File submitted'
zedlmsg = 'The member '||member||' has been submitted to MVS'
'ispexec setmsg msg(isrz000)'
ADDRESS "CMD" "DELETE $$$$$$$$.SPF"
end
EXIT 0
```

**Figure 246: SUBmit REXX for SPF/PC**

## 14.3 Submitting Jobs from Unix

Submitting jobs from Unix can be done in two different ways, described in detail in the next sections:

- Using a Perl script
- Using the netcat program

### *14.3.1 Using a Perl script*

Malcolm Beattie has provided a simple Perl script which can submit jobs using either TCP/IP or Unix Domain Sockets. The script is invoked using one of the following command formats:

```
hercsub 192.168.1.102:3505 dummy.jcl
hercsub /tmp/hercrdr.00C dummy.jcl
```

Here is the Hercsub script:

```
============== hercsub ================
#!/usr/bin/perl
use Socket;
use IO::Socket::UNIX;
use IO::Socket::INET;

if (@ARGV < 1) {
  print STDERR "Usage: hercsub socket_spec [job]\n";
  exit 2;
}

my $spec = shift @ARGV;
my $sock;

if ($spec =~ m{^/}) {
  $sock = IO::Socket::UNIX->new(Peer => $spec);
} else {
  $sock = IO::Socket::INET->new(PeerAddr => $spec);
}

die "Failed to connect to socket $spec: $!\n" unless defined($sock);

while (<>) {
  print $sock $_;
}
========== end of hercsub =============
```

**Figure 247: The Hercsub Perl Script**

## *14.3.2 Using the netcat program*

The netcat (nc) program can also be used to submit files to a Hercules reader via TCP/IP. Install netcat (which is useful for many other things as well) and use the following syntax:

```
nc –w1 localhost 1234 < dummy.jcl
```

For more information on the netcat program, type *man nc* from the Unix shell.

# Appendix A: Supported DASD Device Types

The following tables show the supported DASD device types and models with their sizes. The symbol “[*]” in the size column means that any size can be specified, else the size defaults to the first listed model.

## CKD Devices

| Devicetype-Model | Cylinders | Alternate Cylinders |
|---|---|---|
| IBM 2311 | [*] | |
| IBM 2311-1 | 200 | 2 |
| IBM 2314 | [*] | |
| IBM 2314 | 200 | 3 |
| IBM 3330 | [*] | |
| IBM 3330-1 | 404 | 7 |
| IBM 3330-2 | 808 | 7 |
| IBM 3330-11 | 808 | 7 |
| IBM 3340 | [*] | |
| IBM 3340-1 | 348 | 1 |
| IBM 3340-35 | 348 | 1 |
| IBM 3340-2 | 696 | 2 |
| IBM 3340-70 | 696 | 2 |
| IBM 3350 | [*] | |
| IBM 3350-1 | 555 | 5 |
| IBM 3375 | [*] | |
| IBM 3375-1 | 959 | 1 |
| IBM 3380 | [*] | |
| IBM 3380-1 | 885 | 1 |
| IBM 3380-A | 885 | 1 |
| IBM 3380-B | 885 | 1 |

| Devicetype-Model | Cylinders | Alternate Cylinders |
|---|---|---|
| IBM 3380-D | 885 | 1 |
| IBM 3380-J | 885 | 1 |
| IBM 3380-2 | 1770 | 2 |
| IBM 3380-E | 1770 | 2 |
| IBM 3380-3 | 2665 | 3 |
| IBM 3380-K | 2665 | 3 |
| EMC 3380 K+ | 3339 | 3 |
| EMC 3380 K++ | 3993 | 3 |
| IBM 3390 | [*] | 1 |
| IBM 3390-1 | 1113 | 1 |
| IBM 3390-2 | 2226 | 2 |
| IBM 3390-3 | 3339 | 1 |
| IBM 3390-9 | 10017 | 3 |
| IBM 3390-27 | 32760 | 3 |
| IBM 3390-54 | 65520 | 3 |
| IBM 9345 | [*] | |
| IBM 9345-1 | 1440 | 0 |
| IBM 9345-2 | 2156 | 0 |

**Table 23: Supported CKD DASD Devices**

## FBA Devices

| Devicetype-Model | Blocks |
|---|---|
| IBM 3310 | [*] |
| IBM 3310-1 | 125664 |
| IBM 3370 | [*] |
| IBM 3370-A1 | 558000 |
| IBM 3370-B1 | 558000 |
| IBM 3370-A2 | 712752 |
| IBM 3370-B2 | 712752 |
| IBM 9313 | [*] |
| IBM 9313-1 | 246240 |
| IBM 9332 | [*] |
| IBM 9332-200 | 360036 |
| IBM 9332-400 | 360036 |
| IBM 9336-600 | 554800 |
| IBM 9335 | [*] |
| IBM 9335-1 | 804714 |
| IBM 9336 | [*] |
| IBM 9336-10 | 920115 |
| IBM 9336-20 | 1672881 |
| IBM 9336-25 | 1672881 |
| IBM 0671-08 | 513072 |
| IBM 0671 | 574560 |
| IBM 0671-04 | 624456 |

**Table 24: Supported FBA DASD Devices**

# Appendix B. Configuration of Emulated CPUs

This appendix describes the cooperation of the configuration statements related to the emulated CPUs in more detail. The configuration statements that define the emulated CPUs are:

- ENGINES (Processor engine type)
- MAXCPU (Maximum number of CPUs)
- NUMCPU (Number of emulated CPUs)

## B.1 General Explanations and Rules

The ENGINES parameter specifies the type of engine for each emulated processor (valid types are CP, AP, IP or IL). MAXCPU specifies the maximum number of installed processor engines, whereas NUMCPU defines the number of emulated processor engines that are configured online at IML time.

The ENGINES statement specifies any mixture of CPU types up to MAXCPU, the maximum number of installed processors. NUMCPU configures the processor engines in the order that they are specified on the ENGINES statement.

NUMCPU must be less than or equal to MAXCPU. If NUMCPU is larger than MAXCPU then an error message is issued during the processing of the Hercules configuration file. If it is less than MAXCPU then the remaining engines can be configured online later by the operating system.

## B.1 Examples

Some examples show the cooperation of these statements and the resulting configurations. In the tables the following abbreviations and colours are used:

**CPUs and status**

| | |
|---|---|
| **CP** | CPU of type CP specified |
| **IP** | CPU of type IP specified |
| **AP** | CPU of type AP specified |
| **(D)** | CPU got its type by default |
| **O** | CPU is taken online |
| **F** | CPU is taken offline |

**Colours**

| | |
|---|---|
| xx | Green - Resulting configuration as expected |
| xx | Yellow - Resulting configuration may or may be not as expected (defaults or ignored engines) |
| xx | Red - Resulting configuration is in error. The Hercules configuration file must be corrected. |

**Example 1:**

```
ENGINES CP,CP,CP,AP,IP,IP,CP,CP
MAXCPU 8
NUMCPU 8
```

All CPUs are configured online with the type defined.

| CPU and State | CPU 1 | CPU 2 | CPU 3 | CPU 4 | CPU 5 | CPU 6 | CPU 7 | CPU 8 |
|---|---|---|---|---|---|---|---|---|
| MAXCPU | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| NUMCPU | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| ENGINES | CP | CP | CP | AP | IP | IP | CP | CP |
| | | | | | | | | |
| CPU type | CP | CP | CP | AP | IP | IP | CP | CP |
| Status | O | O | O | O | O | O | O | O |

**Table 25: Correct CPU configuration (example 1)**

**Example 2:**

```
ENGINES CP,CP,CP,AP,IP,IP,CP,CP
MAXCPU 8
NUMCPU 4
```

All CPUs are defined with the specified type and the first four CPUs are taken online. The rest of the CPUs are taken offline.

| CPU and State | CPU 1 | CPU 2 | CPU 3 | CPU 4 | CPU 5 | CPU 6 | CPU 7 | CPU 8 |
|---|---|---|---|---|---|---|---|---|
| MAXCPU | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| NUMCPU | 1 | 2 | 3 | 4 | --- | --- | --- | --- |
| ENGINES | CP | CP | CP | AP | IP | IP | CP | CP |
| | | | | | | | | |
| CPU type | CP | CP | CP | AP | IP | IP | CP | CP |
| Status | O | O | O | O | F | F | F | F |

**Table 26: Correct CPU configuration (example 2)**

**Example 3:**

```
ENGINES CP,CP,CP,AP,IP,IP,CP,CP
MAXCPU 6
NUMCPU 4
```

All CPUs up to the number of MAXCPU are defined with their specified type. Four CPUs, as specified in NUMCPU, are taken online, the other two CPUs are taken offline. Please note that the two excess CPUs that are specified in the ENGINES are ignored, because MAXCPU is set only to six engines.

| CPU and State | CPU 1 | CPU 2 | CPU 3 | CPU 4 | CPU 5 | CPU 6 | CPU 7 | CPU 8 |
|---|---|---|---|---|---|---|---|---|
| MAXCPU | 1 | 2 | 3 | 4 | 5 | 6 | --- | --- |
| NUMCPU | 1 | 2 | 3 | 4 | --- | --- | --- | --- |
| ENGINES | CP | CP | CP | AP | IP | IP | CP | CP |
| | | | | | | | | |
| CPU type | CP | CP | CP | AP | IP | IP | --- | --- |
| Status | O | O | O | O | F | F | --- | --- |

**Table 27: Correct CPU configuration (example 3)**

**Example 4:**

```
ENGINES CP,CP,CP,AP,IP,IP
MAXCPU 8
NUMCPU 4
```

The first four engines are defined with their specified type and are taken online because of the NUMCPU configuration statement. The next two CPUs are also defined with their specified type, but are taken offline, because of the NUMCPU value. MAXCPU is set to eight CPUs but ENGINES specifies only the type of six of the engines, so the remaining two CPUs are defined per default as type CP and are taken offline as specified in NUMCPU.

| CPU and State | CPU 1 | CPU 2 | CPU 3 | CPU 4 | CPU 5 | CPU 6 | CPU 7 | CPU 8 |
|---|---|---|---|---|---|---|---|---|
| MAXCPU | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| NUMCPU | 1 | 2 | 3 | 4 | --- | --- | --- | --- |
| ENGINES | CP | CP | CP | AP | IP | IP | --- | --- |
| | | | | | | | | |
| CPU type | CP | CP | CP | AP | IP | IP | CP(D) | CP(D) |
| Status | O | O | O | O | F | F | F | F |

**Table 28: Correct CPU configuration (example 4)**

**Example 5:**

```
ENGINES CP,CP,AP,IP
MAXCPU 6
NUMCPU 6
```

The first four engines are defined with their specified type and are taken online. The next two CPUs are defined per default as type CP, because the number in MAXCPU exceeds the number of CPUs in the ENGINES statement. These additional two CPUs are also taken online through the NUMCPU statement.

| CPU and State | CPU 1 | CPU 2 | CPU 3 | CPU 4 | CPU 5 | CPU 6 | CPU 7 | CPU 8 |
|---|---|---|---|---|---|---|---|---|
| MAXCPU | 1 | 2 | 3 | 4 | 5 | 6 | --- | --- |
| NUMCPU | 1 | 2 | 3 | 4 | 5 | 6 | --- | --- |
| ENGINES | CP | CP | AP | IP | --- | --- | --- | --- |
| | | | | | | | | |
| CPU type | CP | CP | AP | IP | CP(D) | CP(D) | --- | --- |
| Status | O | O | O | O | O | O | --- | --- |

**Table 29: Correct CPU configuration (example 5)**

**Example 6:**

```
ENGINES CP,CP,AP,IP
MAXCPU 6
NUMCPU 8
```

This configuration leads to an error. While the first four CPUs would be taken online with their specified types and the next two CPUs would be of type CP per default and be taken offline, the excess engines from NUMCPU compared against MAXCPU lead to an error message and a failing configuration.

| CPU and State | CPU 1 | CPU 2 | CPU 3 | CPU 4 | CPU 5 | CPU 6 | CPU 7 | CPU 8 |
|---|---|---|---|---|---|---|---|---|
| MAXCPU | 1 | 2 | 3 | 4 | 5 | 6 | --- | --- |
| NUMCPU | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| ENGINES | CP | CP | AP | IP | --- | --- | --- | --- |
| | | | | | | | | |
| CPU type | CP | CP | AP | IP | CP(D) | CP(D) | CP(D) | CP(D) |
| Status | O | O | O | O | O | O | F | F |

**Table 30: Incorrect CPU configuration (example 6)**

# Appendix C. Syntax

This book uses two kinds of describing the syntax of configuration statements, console commands and utilities.These are:

- Syntax descriptions
- Syntax diagrams

## C.1 Reading Syntax Descriptions

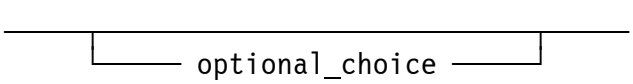
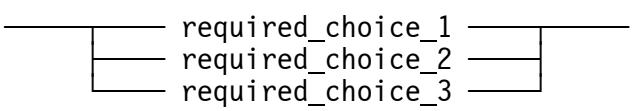
All syntax descriptions in this book (configuration statements, console commands and utilites) use a common structure as described in the following table.

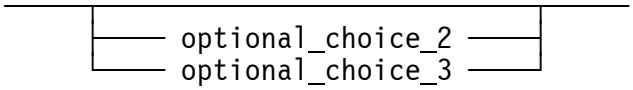
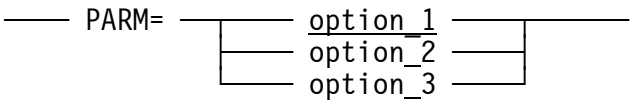
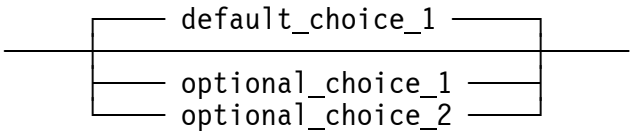
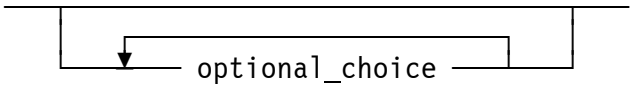
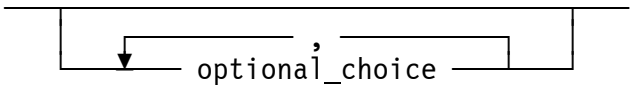
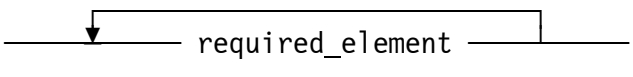
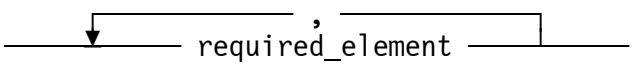
| Syntax Element | Description |
|---|---|
| **KEYWORDS** | Keywords are denoted with upper case letters. Obey the spelling. In the actual statements or commands they can be coded in upper case or lower case letters. |
| ***variables*** | All user defined values are denoted with lower case italic letters. In the actual statements or commands they can be coded in upper case or lower case letters. |
| **{ }** | Signifies that all, or some portion, of the code elements between the braces are required elements. Note that the braces are not part of the statements and must be not coded. |
| **[ ]** | Signifies that all, or some portion of the code elements between the square brackets can optionally appear but are not required elements. Note that the square brackets are not part of the statements and must be not coded. |
| **\|** | The OR symbol signifies that you may use only one of the code elements or values from the possible choices. Note that the OR symbol is not part of the statements and must be not coded. |
| **xxx,...** | Signifies that there can be more than one value in a comma delimited list. Note that the dots are not part of the statements and must be not coded. |
| **xxx ...** | Signifies that there can be more than one value in a blank space delimited list. Note that the dots are not part of the statements and must be not coded. |

**Table 31: Reading Syntax Descriptions**

## C.2 Reading Syntax Diagrams

All syntax diagrams in this book (configuration statements, console commands and utilities) use a common structure as described in the following table.

| Symbol | Description |
|---|---|
|  | This symbol indicates the beginning of a syntax diagram. |
|  | This symbol indicates the end of a syntax diagram. |
|  | This symbol indicates that the syntax diagram is continued on the next line. |
|  | This symbol indicates that the syntax diagram is a continuation from the previous line. |
|  | A required element (keyword or variable) appears on the main path of the horizontal line. You must specify this element. |
|  | An optional element (keyword or variable) appears below the main path of the horizontal line. You may or may not specify this element. |
|  | A required choice (keyword or variable) appears vertically stacked in the main path of the horizontal line. You must choose one of the available options. |

| Symbol | Description |
|---|---|
|  | An optional choice (keyword or variable) appears vertically stacked below the main path of the horizontal line. You may choose one of the available options. |
|  | A keyword with options. Only one of the available options may be specified. The underscored option is the default if the whole keyword statement is not coded. |
|  | An optional choice (keyword or variable) with default appears vertically stacked with the default value above the main path of the horizontal line and the remaining optional elements below the main path of the horizontal line. Only one of the available options may be specified. If none of these elements is explicitly specified, the default above the main line is taken. |
|  <br>  | An arrow returning to the left of an element below the main path of the horizontal line indicates an optional repeatable item. A character within the arrow path means that repeated items have to be separated by that character. If there is no character within the arrow path then the items are separated by a blank. |
|  <br>  | An arrow returning to the left of an element on the main path of the horizontal line indicates an required repeatable item. A character within the arrow path means that repeated items have to be separated by that character. If there is no character within the arrow path then the items are separated by a blank. |

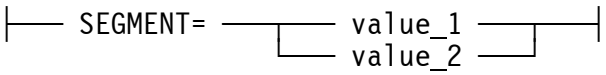
| Symbol | Description |
|---|---|
| | This symbol is a reference to a syntax segment, which is described separately below the main syntax diagram. Complex syntax diagrams are occasionally broken into separated simpler segments. |
|  | This symbol indicates a syntax segment which is referenced from a main syntax diagram that is shown above the syntax segment. |
| KEYWORDS | Keywords are denoted with upper case letters. Obey the spelling. Lower case letters are optional and can be omitted (for example DISable). In the actual statements or commands the keywords can be coded in upper case or lower case letters. |
| *variables* | All user defined values are denoted with lower case italic letters. They represent user supplied names or values. In the actual statements or commands they can be coded in upper case or lower case letters. |

**Table 32: Reading Syntax Diagrams**

## C.3 Sample Syntax Description

The following figure shows a sample of a complex syntax description. This is not an example of an existing system parameter or panel command. It is used mainly to demonstrate the “look and feel” of syntax descriptions.

**Syntax**

```
CMDNAME required_argument [optional_argument]
        {required_choice_1 | required_choice_2 | required_choice_3}
        [optional_choice_1 | optional_choice_2]
        REQUIRED_KEYWORD=variable
        [OPTIONAL_KEYWORD=variable]
        [DEFAULT_KEYWORD | KEYWORD_1 | KEYWORD_2]
        [KEYWORD=default_choice | KEYWORD={choice_1 | choice_2}]
```

*repeatable_item_n* {*repeatable_item_n* ...}

*repeatable_item_n* {,*repeatable_item_n* ,...}

fragment_name

(*variable_1*, *variable_2*)

**fragment_name**

[DEFAULT KEYWORD | KEYWORD_1 | KEYWORD_2 | KEYWORD_3 | KEYWORD_4]

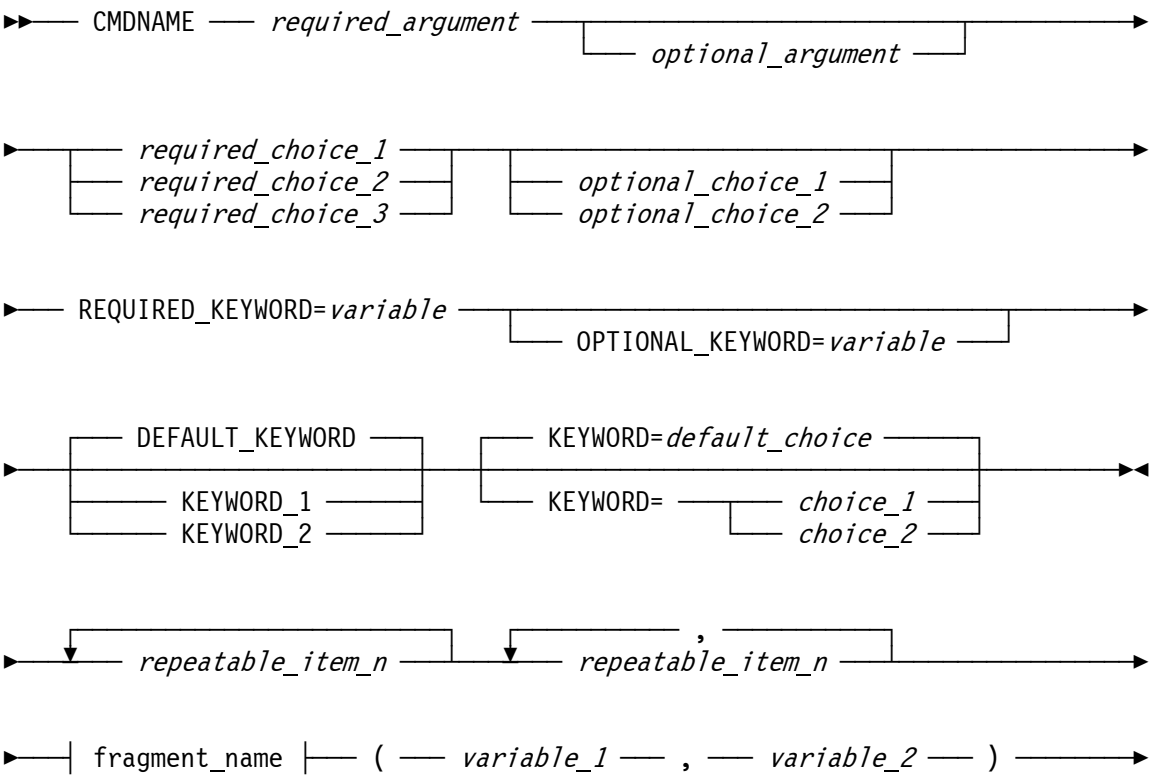
KEYWORD={*choice_1* | *choice_2* | *choice_3* | *choice_4*}

**Figure 248: Sample Syntax Description**

## C.4 Sample Syntax Diagram

The next figure shows a sample of a complex syntax diagram. It shows the same example as in the syntax description in the previous section. Like in the example before it is not based on an existing system parameter or panel command. It is used mainly to demonstrate the “look and feel” of syntax descriptions.

**Syntax**



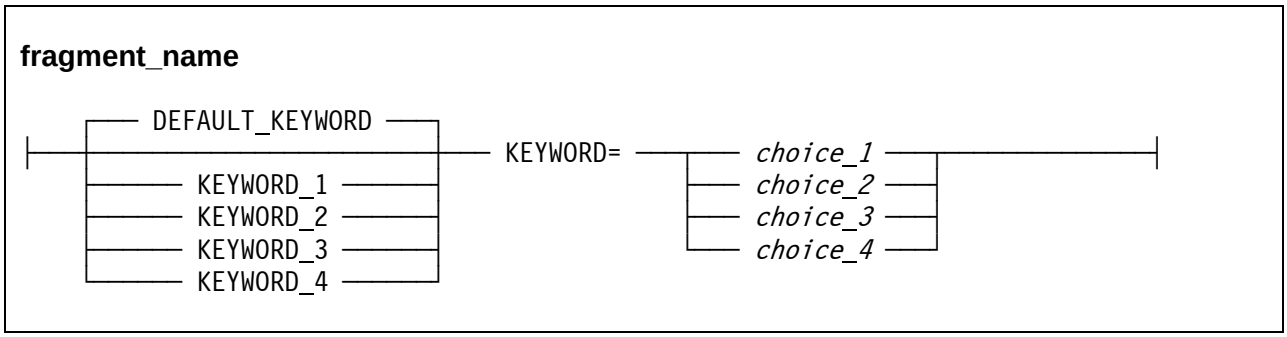



**Figure 249: Sample Syntax Diagram**

# Appendix D. Links

- **The Hercules System/370, ESA/390, and z/Architecture Emulator**

  http://www.hercules-390.eu

- **Hercules source code repositories**

  https://github.com/rbowler/spinhawk (release 3.xx development stream)

  https://github.com/rbowler/sandhawk (release 4.xx development stream)

  https://github.com/hercules-390/hyperion (cutting-edge developer sandbox)

- **Hercules Developer Snapshots (Dave Wade)**

  http://www.smrcc.org.uk/members/g4ugm/snapshots/

- **Hercules PDF Documentation (Peter Glanzmann)**

  http://hercdoc.glanzmann.org

- **The MVS Tur(n)key System, Version 3 (Volker Bandke)**

  http://www.bsp-gmbh.com/turnkey/index.html

- **Hercules WinGUI (“Fish”, David B. Trout)**

  http://www.softdevlabs.com/Hercules/hercgui-index.html

- **CTCI-WIN (“Fish”, David B. Trout)**

  http://www.softdevlabs.com/Hercules/CTCI-WIN-index.html

- **Hercules Studio (Jacob Dekel)**

  http://www.mvsdasd.org/hercstudio

- **Hebe – Hercules Image Manager (Robin Atwood)**

  http://kde-apps.org/content/show.php/Hebe?content=126738

- **WinPcap, Politecnico di Torino**

  http://www.winpcap.org

- **Vista tn3270, Tom Brennan Software**

  http://www.tombrennansoftware.com

- **X3270, Paul Mattes**

  http://x3270.bgp.nu

- **AWSBROWSE (“Fish”, David B. Trout)**

  http://www.softdevlabs.com/Hercules/hercgui-index.html

- **XMIT Manager**

  www.cbttape.org

- **CBT MVS Utilities Tape (CBTTAPE)**

  www.cbttape.org

- **Microsoft Visual C++ 2008 Express**

  http://www.microsoft.com/express/download/

- **ZLIB**

  http://www.zlib.net

  http://www.softdevlabs.com/Hercules/ZLIB1-1.2.3-bin-lib-inc-vc2008-x86-x64.zip

- **BZIP2**

  http://www.bzip.org

  http://www.softdevlabs.com/Hercules/BZIP2-1.0.5-bin-lib-inc-vc2008-x86-x64.zip

- **PCRE**

  http://www.pcre.org

  http://www.softdevlabs.com/Hercules/PCRE-6.4.1-bin-lib-inc-vc2008-x86-x64.zip

Hercules Emulator



Hercules System/370, ESA/390,
z/Architecture Emulator

# User Reference Guide

Version 3 Release 11

HEUR031100-01